

MODIS Level 1A/Geolocation Software Requirements Specification



September 11, 1997

SDST-059A

**MODIS
Level 1A/Geolocation
Software Requirements Specification**

Prepared by:

Keith Degnan, SAIC/GSC Lead Instrument Analyst	Date
---	------

Frederick Patt, SAIC/GSC SDST System Analyst	Date
---	------

Reviewed by:

Jeffrey Blanchette, SAIC/GSC Lead Geolocation Analyst	Date
--	------

James Firestone, SAIC/GSC SDST STTG Lead	Date
---	------

Dr. Al Fleig, University of Maryland MODIS Data System Scientist	Date
---	------

Barbara Putney, GSFC/Code 920 MODIS Systems Engineer	Date
---	------

Laurie Schneider, SAIC/GSC SDST R&QA Manager	Date
---	------

Richard Isaacman, SAIC/GSC MODIS SDST Task Leader	Date
--	------

Approved By:

Edward Masuoka, GSFC/Code 920.2 MODIS SDST Manager	Date
---	------

MODIS Level 1A/Geolocation Software Requirements Specification

Table of Contents

1. INTRODUCTION	1-1
1.1 Identification.....	1-1
1.2 Scope.....	1-1
1.3 Purpose and Objectives.....	1-1
1.4 Volume Status and Schedule.....	1-1
1.5 Document Organization.....	1-1
2. RELATED DOCUMENTATION	2-1
2.1 Parent Documents.....	2-1
2.2 Related Documents.....	2-1
3. REQUIREMENTS APPROACH	3-1
4. EXTERNAL INTERFACE DESCRIPTION	4-1
4.1 External Interface Requirements for Level 1A Software.....	4-1
4.1.1 Level0_Product.....	4-1
4.1.2 Level1A_Parameter_Set.....	4-1
4.1.3 Level1A_Product.....	4-2
4.1.4 Log_messages.....	4-2
4.2 External Interface Descriptions for Geolocation Software.....	4-2
4.2.1 Level 1A Product.....	4-2
4.2.2 Geolocation Parameter Set.....	4-3
4.2.3 Spacecraft Information.....	4-3
4.2.4 Digital Elevation Model.....	4-3
4.2.5 Level 1B Product.....	4-4
4.2.6 Level 2 Products.....	4-4
4.2.7 Control Point Library.....	4-4
4.2.8 Geolocation Product.....	4-4
4.2.9 Quality Assurance/Quality Control Results.....	4-5
4.2.10 Control Point History.....	4-5
5. REQUIREMENTS SPECIFICATION	5-1
5.1 Process and Data Requirements.....	5-1
5.1.1 Level 1A Process and Data Requirements.....	5-2
5.1.2 Geolocation Process and Data Requirements.....	5-9
5.2 Performance and Quality Engineering Requirements.....	5-25
5.3 Safety Requirements.....	5-25
5.4 Security and Privacy Requirements.....	5-25
5.5 Implementation Constraints.....	5-26
5.6 Site Adaptation.....	5-26
5.7 Design Goals.....	5-27

6. REQUIREMENTS TRACEABILITY	6-1
7. PARTITIONING FOR RELEASES	7-1
8. GLOSSARY/SYMBOLS	8-1
APPENDIX A: ACRONYMS	A-1
APPENDIX B: LEVEL 1A DATA FLOW DIAGRAMS	B-1
APPENDIX C: LEVEL 1A DATA DICTIONARY	C-1
C.1 Alphabetic Listing	C-1
C.2 Major Structures Decomposed.....	C-14
C.2.1 Level0_data	C-14
C.2.2 Level1A_data	C-21
C.2.3 Level1A_Parameters.....	C-23
C.2.4 Log_messages.....	C-23
APPENDIX D: GEOLOCATION DATA FLOW DIAGRAMS	D-1
APPENDIX E: GEOLOCATION DATA DICTIONARY	E-1
E.1 Alphabetic Listing	E-1
E.2 Major Structures Decomposed.....	E-24
E.2.1 Level_1A_Data.....	E-24
E.2.2 Geolocation Parameters.....	E-26
E.2.3 level1a_conversion_and_validation_parameters	E-29
E.2.4 control_point_parameters.....	E-30
E.2.5 Spacecraft_Data	E-31
E.2.6 Elevation_Data	E-32
E.2.7 Control_Points.....	E-32
E.2.8 Geolocation_Data	E-33
E.2.9 Control_Point_Matchups	E-42
E.2.10 Level_1B_Data.....	E-42
E.2.11 Level_2_Data	E-44

List of Figures

Figure 4-1. Level 1A Context Diagram	4-1
Figure 4-2. Geolocation Context Diagram.....	4-3
Figure B-1. MODIS Level 1A Processing.....	B-1
Figure B-2. Initialize_Level_1A_processing.....	B-2
Figure B-3. Process_Packets.....	B-3
Figure B-4. Verify_MODIS_packet.....	B-4
Figure B-5. Create_MODIS_Scans.....	B-5
Figure B-6. Create_decommutated_eng_mem_list.....	B-6
Figure D-1. MODIS Geolocation.....	D-1
Figure D-2. Process Level 1A Data.....	D-2
Figure D-3. Compute Geolocation Fields.....	D-3
Figure D-4. Determine Instrument Lines of Sight.....	D-4
Figure D-5. Determine Observed Locations	D-5
Figure D-6. Correct Observed Locations.....	D-6
Figure D-7. Compute Sensor and Solar Vectors.....	D-7
Figure D-8. Compute Solar Diffuser Angles.....	D-8
Figure D-9. Compute Lunar Vectors	D-9
Figure D-10. Write Geolocation Product.....	D-10
Figure D-11. Assess Geolocation Accuracy	D-11

List of Tables

Table 6-1. L1A/Geolocation Requirements Traceability	6-1
Table B-1. Control Activation.....	B-1

MODIS Level 1A/Geolocation Software Requirements Specification

1. INTRODUCTION

1.1 Identification

This is the requirements specification for the Moderate Resolution Imaging Spectroradiometer (MODIS) Level 1A (L1A)/Geolocation software.

1.2 Scope

This document identifies the requirements that must be met by the L1A/Geolocation software. The requirements methodology is also briefly described, and design goals are discussed. All software designers, programmers, and testers of the L1A/Geolocation software should be familiar with the content of this document. Additional parties, such as Quality Assurance (QA) personnel, customers of L1A/Geolocation data, and Earth Observing System (EOS) Data and Information System (EOSDIS) personnel, may be interested in portions of this document.

1.3 Purpose and Objectives

The purpose of this document is to specify the requirements of the MODIS L1A/Geolocation software. The objective of this document is to serve as the baseline for software development and testing of the L1A/Geolocation software.

1.4 Volume Status and Schedule

This document replaces the previous MODIS L1A requirements document (MODIS Technical Report Series, Volume 1, MODIS Level 1A Software Baseline Requirements, NASA Technical Memorandum (TM) 104594, Vol. 1, April 1994). Changes to the L1A/Geolocation requirements in the future will be approved through a configuration control process; it is anticipated that this document will be updated for future releases of the software.

1.5 Document Organization

This document is organized following the NASA-2100 Software Documentation Standard NASA-DID-P200, Requirements with several small changes. The organization of NASA-2100 mandates the following sections; tailoring of the standard is noted in brackets on the descriptions.

- Section 1: Introduction - Identifies the document and describes its purpose.
- Section 2: Related Documentation - Includes MODIS programmatic documents, Hughes Santa Barbara Remote Sensing (SBRs) Center documents on the MODIS instrument, and EOSDIS Core System (ECS) documents.

- Section 3: Requirements Approach - Describes the overall approach used to develop the requirements presented in this document.
- Section 4: External Interface Description - Describes the interfaces between the software and its external environment. [This section is External Interface Requirements in the standard.]
- Section 5: Requirements Specification - Provides the detailed requirements for the software.
- Section 6: Requirements Traceability - Provides the mapping between the requirements described in this document and higher-level requirements.
- Section 7: Partitioning for Releases - Describes how the software will differ between each software version.
- Section 8: Abbreviations and Acronyms [moved to Appendix A]
- Section 9: Glossary/Symbols [added Symbols and renumbered section]
- Section 10: Notes [removed]
- Section 11: Appendices [section heading removed and each appendix is lettered as follows:
 - Appendix A: Abbreviations and Acronyms [Moved to Appendix A]
 - Appendix B: Level 1A Data Flow Diagrams
 - Appendix C: Level 1A Data Dictionary
 - Appendix D: Geolocation Data Flow Diagrams
 - Appendix E: Geolocation Data Dictionary]

2. RELATED DOCUMENTATION

2.1 Parent Documents

The complete MODIS Science Data Support Team (SDST) Documentation Tree can be found in Figure 1-1 of the MODIS Science Data Processing Software Version 1 Requirements Specification.

Specifically, this document has as its parent documents:

- MODIS Data Management Plan; SDST-006; October 25, 1995.
- MODIS Software Management Plan; SDST-002; October 24, 1995.
- MODIS Science Data Processing Software Requirements Specification Version 2 and Beyond, SDST-089, December 12, 1996.
- MODIS Level 1A Earth Location: Algorithm Theoretical Basis Document Version 3.0, R. Wolfe, et al., April 25, 1997.
- MODIS Version 2 Science Computing Facility Software Delivery Guide, SDST-096, May 5, 1997.
- Team Leader Working Agreement for MODIS Between EOS AM & PM Projects GSFC and the MODIS Science Team Leader; GSFC 421-12-14-02; April 1994.

2.2 Related Documents

The following documents provide information relevant to this document:

- EOS General Instrument Interface Specification (GIIS) for the EOS Observatory, GSFC 420-03-02, December 1992.
- MODIS Technical Report Series, Volume 1, MODIS Level 1A Software Requirements, NASA Technical Memorandum 104594, Vol. 1, April 1994.
- MODIS Program, Engineering Telemetry Description, Contract Document Requirements List (CDRL) 305, April 1994.
- MODIS Level 1A Earth Location: Algorithm Theoretical Basis Document Version 2.0, R. Wolfe, et al., April 1995.
- NASA Software Documentation Standard, Software Engineering Program, NASA-STD-2100-91, July 29, 1991.
- Structured Analysis and System Specification, T. DeMarco, Yourdon Press, NY, 1978.
- Structured Development of Real-Time Systems, Volumes 1-3, P. Ward and S. Mellor, Yourdon Press, NY, 1986.

(This page intentionally left blank.)

3. REQUIREMENTS APPROACH

The MODIS L1A/Geolocation requirements for Version 1 (V1) were developed by performing two basic activities:

1. Collecting high-level requirements, and
2. Modeling these requirements to develop design-to specifications.

Each of these activities will be briefly discussed in this section.

Several sources of high-level requirements were used as the basis of this requirements phase. Parent documents were reviewed to ensure that any applicable project requirements flowed down to the software requirements. Previous requirements work done for the MODIS L1A/Geolocation software (NASA TM 104594, Vol. 1) was checked for any additional high-level requirements that might have been missed when the parent documents were reviewed. Lessons learned during the Beta software development phase were also considered when these V1 requirements were collected. Finally, the domain knowledge of the L1A and Geolocation software analysts contributed to the high-level view of the software requirements.

After the high-level requirements were collected, a requirements modeling effort was undertaken to specify the requirements for software development. The methodology used to create the requirements model is based on DeMarco's structured analysis (Structured Analysis and System Specification, 1979), with real-time extensions defined by Ward and Mellor (Structured Development of Real-Time Systems, 1986). This methodology combines both graphical and textual elements that, when taken together, form a model of the functional decomposition of the software, including both process and data detail. The MODIS SDST has selected the Computer-Aided Software Engineering (CASE) tool *Teamwork*, produced by Cadre Technologies Inc., as the modeling tool for use by SDST when developing the L1A and Geolocation software. *Teamwork* Structured Analysis with Real-Time (SA/RT) extensions provides interfaces for creating Data Flow Diagrams (DFDs), Data Dictionary Entries (DDEs), and process specifications (p-specs), as well as utilities for balancing various elements of the constructed requirements model.

For V1 of the software, separate requirements models for L1A and Geolocation software were developed. This was done for several reasons:

- The MODIS project plans to have separate archive products for L1A and Geolocation data;
- It is likely that the Geolocation software will be re-run at times when the L1A software would not need to be re-run;
- The Beta Version of the software had separate executables, and the V1 software will also be developed as separate executables.

The organization of the both the external interface requirements and requirements specification sections of this document reflects the separation of the L1A and Geolocation requirements models.

(This page intentionally left blank.)

4. EXTERNAL INTERFACE DESCRIPTION

4.1 External Interface Requirements for Level 1A Software

The context diagram for L1A (Figure 4-1) shows the external elements that interact with the L1A software. See Section 8, Glossary/Symbols, for a brief description of the notation.

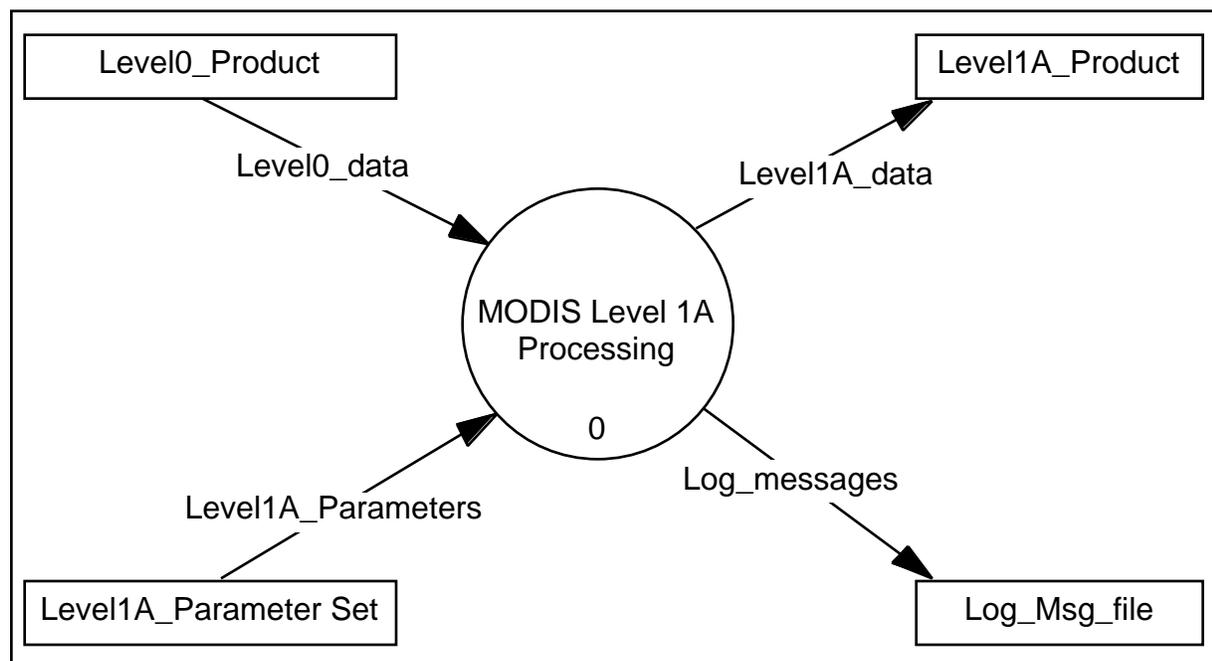


Figure 4-1. Level 1A Context Diagram

4.1.1 Level0_Product

The Level 0 product, produced by the EOS Data and Operations System (EDOS), contains the raw MODIS instrument data as output by the instrument itself, as well as product metadata. The data is formatted in accordance with the Consultative Committee for Space Data Systems (CCSDS) packet concept, with the packet contents and format defined by Hughes SBRS (CDRL 305).

The exact structure of the EDOS-produced Level 0 products is still To Be Determined (TBD); however, draft materials provided by EDOS as well as discussions held among MODIS, EOS-AM, and EDOS project staff indicate that the Level 0 process will properly order the MODIS packets, and that the Level 0 product will contain metadata which will allow proper validation of the Level 0 data.

4.1.2 Level1A_Parameter_Set

Only two L1A parameters are identified as needing to be supplied externally. These are processing mode (production or expedited) and spacecraft ID (EOS-AM1, EOS-PM1, etc).

4.1.3 Level1A_Product

This is the primary output of the L1A process. It contains:

- Unconverted radiances for all detectors at all sources;
- Decommuted instrument engineering and memory information;
- Spacecraft ancillary data;
- EOS standard metadata;
- Granule-specific metadata;
- Data quality information.

The exact file format for V1 will be described in the L1A V1 product format specification.

4.1.4 Log_messages

The L1A software will produce status messages when warning or error conditions are identified. These messages will be stored in a log file by using SDPTK routines, and the resulting file will be shipped to the TLCF for use by the L1A analyst.

4.2 External Interface Descriptions for Geolocation Software

4.2.1 Level 1A Product

The L1A product (MOD01) contains the granule-specific information which is needed to perform geolocation processing for the granule. This information consists of:

- Granule-level metadata, including number of scans, orbit number and granule number within the orbit;
- Earth sector start time, mirror side, and number of Earth sector frames for each scan;
- Raw scan mirror encoder data and Earth sector mirror start positions; and
- Raw spacecraft ancillary data.

The start times, mirror sides and numbers of frames will have been unpacked and converted. The mirror encoder data, mirror start positions, spacecraft ancillary data and radiances will be unpacked but not converted and stored as separate objects in the L1A product.

The context diagram for Geolocation (Figure 4-2) shows the external elements that interact with the Geolocation software.

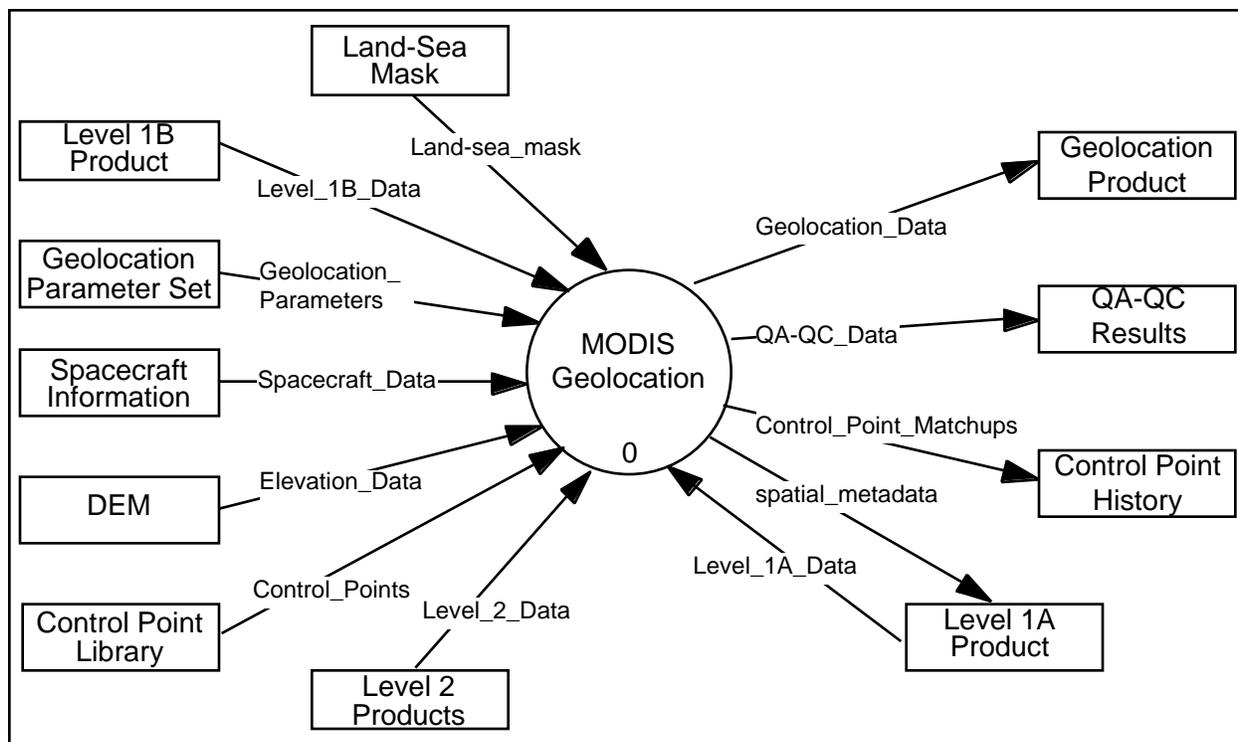


Figure 4-2. Geolocation Context Diagram

4.2.2 Geolocation Parameter Set

The geolocation processing requires a large parameter set. The values of the parameters will not be granule dependent. The parameters can be grouped into the following major categories: instrument model (as specified in the Geolocation Algorithm Theoretical Basis Document [ATBD]); L1A conversion and validation; and control point parameters.

The geolocation parameter set is implemented and maintained external to the geolocation software, to enable parameter values to be updated without the need to rebuild the software.

4.2.3 Spacecraft Information

The ECS tools have the capability to access spacecraft orbit and attitude information which will be provided independently of the MODIS data stream. The geolocation software will have the option to access the data from this source in place of the embedded spacecraft ancillary data through the Science Data Production Toolkit (SDPTK).

4.2.4 Digital Elevation Model

The geolocation process is required to correct the observed locations for the effect of terrain height to achieve the accuracy specified in the V1 System Requirements Specification. The terrain height information will be provided in the form of a Digital

Elevation Model (DEM) file or files, which will contain gridded elevation data of sufficient resolution and accuracy to meet the geolocation accuracy requirements.

The DEM data will be provided by ECS and will be preprocessed by the MODIS SDST to a format and representation which is optimized for geolocation processing.

4.2.5 Level 1B Product

The L1B product (MOD02) contains the granule-specific information which is needed to assess the accuracy of the geolocation processing. This information consists of:

- Calibrated radiance values for the bands used in the control point correlation algorithms.

4.2.6 Level 2 Products

The L2 products (including MOD35) contain granule-specific processed MODIS data which can augment the geolocation process's accuracy assessment. This information consists of:

- Level 2 MODIS Classification Masks
- Other Level 2 products that may contribute to the identification of regions in a granule that should be excluded for the purpose of geolocation process accuracy assessment.

4.2.7 Control Point Library

The control point (CP) library will be used to assess the accuracy of the geolocation processing. The results of control point correlation will be analyzed to determine static or systematic errors and corrections will be incorporated into the geolocation software to ensure that the required accuracy is maintained.

The CP library will be generated by the MODIS SDST. Two types of CPs will be used:

- Land CPs, consisting of clearly visible landmarks in one or more MODIS bands, based on high-resolution data which approximates MODIS bands (e.g., Landsat Thematic Mapper [TM]); and
- Ocean CPs, consisting of islands which are visible at MODIS resolution, based on a high-resolution coastline database (e.g., the World Vector Shoreline).

4.2.8 Geolocation Product

The geolocation product (MOD03) contains the results of geolocation processing which are used by higher-level MODIS processes. The format of the product will be in Hierarchical Data Format (HDF) and will be described in the Geolocation V1 product format specification.

4.2.9 Quality Assurance/Quality Control Results

The geolocation software will produce status messages and reports which document the quality of the input data and any processing problems encountered. These results will be output to external files using Science Data Production (SDP) tools and transmitted to the TLCF to enable overall process monitoring and problem analysis.

4.2.10 Control Point History

The results of the CP correlation processing will be a set of CP matchups for each granule for which CPs are found. These results will be transmitted to the TLCF at the conclusion of the Product Generation Executive (PGE), and maintained there as a CP history, which will be analyzed to determine the static or slowly varying errors.

(This page intentionally left blank.)

5. REQUIREMENTS SPECIFICATION

This section specifies the requirements for the L1A and Geolocation software. The first section is divided into two subsections which specify the process and data requirements for the L1A and Geolocation software, respectively. Section two through six contain the performance and quality engineering requirements, safety requirements, security and privacy requirements, implementation constraints, and site adaptation requirements. Finally, section seven discusses design goals for the software.

Each requirement is prefixed with a mnemonic label of the form PRxx-Y-z.z-n, where:

- PRxx is the MODIS process number (PR01 for L1A, PR03 for Geolocation);
- Y is the type of requirement (F, functional;
P, performance;
Q, quality;
I, Implementation constraints;
S, Site Adaptation);
- z.z is the process number from the requirements model (applies to functional requirements only); and
- n is a counter

Using this convention, the first requirement listed in process 1.2.1 of the L1A requirements would be uniquely identified as PR01-F-1.2.1-1, and the second Implementation Constraint requirement (as it applies to L1A software) would be labeled PR01-I-2. This method provides flexibility to add requirements in the future without renumbering by simply incorporating changes in the requirements model for functional requirements, or by changing the respective section within this document for other requirements.

5.1 Process and Data Requirements

Sections 5.1.1 and 5.1.2 are organized using the same hierarchy as their respective requirements model DFDs. This means that headings within 5.1.1 and 5.1.2 are directly related to the decomposition of functionality shown in the DFDs contained in Appendices B and D. All processes (bubbles on the DFD) show the inputs required and the output produced by the process. For processes that have further decompositions (parent processes), the input/output list is followed by a reference to which diagram illustrates the decomposition. For processes that do not have decompositions (child processes), the input/output list is followed by a list of p-specs. These p-specs form the functional requirements for the software.

5.1.1 Level 1A Process and Data Requirements

PR01-1 Initialize_Level_1A_Processing (Process 1 on DFD 0, Figure B-1)

Input/Output:

Level1A_parameters : data_in
Level0_data : data_in
Init_log_msgs : data_out
MODIS_packets : data_out
Init_metadata : data_out

See DFD 1 (Figure B-2) for decomposition.

PR01-1.1 Open_Level0_File (Process .1 on DFD 1, Figure B-2)

Input/Output:

Level0_data : data_in
MODIS_packets : data_out
Level0_open_metadata : data_out
Level0_open_log_msgs : data_out
Level0_file_header : data_out

Requirements:

- PR01-F-1.1-1. Shall open the Level0_data file.
- PR01-F-1.1-2. Shall abort processing if the Level0_data file open fails.
- PR01-F-1.1-3. Shall write Level0_open_metadata.
- PR01-F-1.1-4. Deleted
- PR01-F-1.1-5. Shall write L0_open_log_msgs to Log_messages when errors occur in the Open_Level0_file process.

PR01-1.2 Verify_Level0_file_header (Process .2 on DFD 1, Figure B-2)

Input/Output:

Level0_file_header : data_in
Level1A_Parameters : data_in
Level0_header_log_msgs : data_out

Requirements:

- PR01-F-1.2-1. Shall verify that the Spacecraft_indicator contained in the Level1A_Parameters is consistent with the Spacecraft_indicator found in the Level 0 file header.
- PR01-F-1.2-2. Shall abort processing if Spacecraft_indicator verification fails.
- PR01-F-1.2-3. Deleted
- PR01-F-1.2-4. Deleted
- PR01-F-1.2-5. Shall write Level0_header_log_msgs to Log_messages if the Spacecraft_indicator verification fails.

PR01-2 Process_Packets (Process 2 on DFD 0, Figure B-1)**Input/Output:**

MODIS_packets : data_in
 Packet_log_msgs : data_out
 Current_HK_telem : data_out
 Prior_HK_telem : data_out
 Sci_eng_data : data_out
 Eng_Mem_data : data_out
 Current_packet_state : data_out
 Unpacked_radiance_packet_data : data_out Static_eng_mem_data : data_out

See DFD 2 (Figure B-3) for decomposition.

PR01-2.1 Read_MODIS_packet (Process .1 on DFD 2, Figure B-3)**Input/Output:**

MODIS_packets : data_in
 Read_packet_log_msgs : data_out
 Program_stop : control_out
 MODIS_pkt : data_out

Requirements:

- PR01-F-2.1-1. Shall read MODIS_packets from the Level0_data file.
- PR01-F-2.1-2. Deleted
- PR01-F-2.1-3. Shall issue Program_stop if end-of-file condition is detected.
- PR01-F-2.1-4. Shall write Read_packet_log_msgs to Log_messages when errors occur in the Read_MODIS_packet process.
- PR01-F-2.1-5. Shall issue Program_stop if MODIS_packets cannot be read from Level0_data file.

PR01-2.2 Verify_MODIS_packet (Process .2 on DFD 2, Figure B-3)**Input/Output:**

MODIS_pkt : data_in
 MODIS_pkt : data_out
 Verify_packet_log_msgs : data_out
 Current_packet_state : data_out
 Abort_current_packet_processing : control_out

See DFD 2.2 (Figure B-4) for decomposition.

PR01-2.2.2 Verify_CCSDS_header_info (Process .2 on DFD 2.2, Figure B-4)**Input/Output:**

MODIS_pkt : data_in
 Verify_CCSDS_log_msgs : data_out
 Abort_current_packet_processing : control_out
 MODIS_pkt : data_out

Requirements:

- PR01-F-2.2.2-1. Shall verify that the values of the CCSDS header fields are consistent with values contained within SBRS CDRL 305.
- PR01-F-2.2.2-2. Shall Abort_current_packet_processing if any CCSDS primary header field fails verification.
- PR01-F-2.2.2-3. Deleted
- PR01-F-2.2.2-4. Shall write Verify_CCSDS_log_msgs to Log_messages when errors occur in the Verify_CCSDS_header_info process.

PR01-2.2.3 Verify_MODIS_header_info (Process .3 on DFD 2.2, Figure B-4)**Input/Output:**

MODIS_pkt : data_in
 Abort_current_packet_processing : control_out
 Verify_MODIS_log_msgs : data_out
 MODIS_pkt : data_out
 Packet_header : data_out

Requirements:

- PR01-F-2.2.3-1. Shall verify that the values of all MODIS header fields are within ranges specified in SBRS CDRL 305.
- PR01-F-2.2.3-2. Shall Abort_current_packet_processing if a MODIS packet fails verification of any MODIS header field.
- PR01-F-2.2.3-3. Deleted
- PR01-F-2.2.3-4. Shall write Verify_MODIS_log_msgs to Log_messages when errors occur in the Verify_MODIS_header_info process.

PR01-2.2.4 Verify_packet_checksum (Process .4 on DFD 2.2, Figure B-4)**Input/Output:**

MODIS_pkt : data_in
 Abort_current_packet_processing : control_out
 Verify_checksum_log_msgs : data_out
 MODIS_pkt : data_out

Requirements:

- PR01-F-2.2.4-1. Shall calculate the packet checksum using an algorithm to be supplied by SBRS.
- PR01-F-2.2.4-2. Shall Abort_current_packet_processing if the calculated packet checksum is not the same as the packet_checksum in the MODIS_pkt.
- PR01-F-2.2.4-3. Deleted
- PR01-F-2.2.4-4. Shall write Verify_checksum_log_msgs to Log_messages when errors occur in the Verify_packet_checksum process.

PR01-2.3 Process_MODIS_packet (Process .3 on DFD 2, Figure B-3)**Input/Output:**

Packet_header : data_in
 MODIS_pkt : data_in
 Abort_current_packet_processing : control_out
 MODIS_pkt : data_out
 Missing_packet_log_msgs : data_out

Requirements:

- PR01-F-2.3-1. Shall determine the type of packet from information in the CCSDS secondary header and the MODIS header.
- PR01-F-2.3-2. Shall determine if the packet belongs in the current scan.
- PR01-F-2.3-3. Shall remove redundant packets.
- PR01-F-2.3-4. Shall determine if packets are missing from the current scan.
- PR01-F-2.3-5. Shall write Missing_packet_log_msgs to Log_messages when missing packets are detected.

Requirements:

- PR01-F-2.3.1-1. Deleted
- PR01-F-2.3.1-2. Deleted
- PR01-F-2.3.2-1. Deleted
- PR01-F-2.3.2-2. Deleted
- PR01-F-2.3.2-3. Deleted
- PR01-F-2.3.2-4. Deleted
- PR01-F-2.3.2-5. Deleted
- PR01-F-2.3.2-6. Deleted
- PR01-F-2.3.2-7. Deleted

PR01-2.4 Unpack_packet_contents (Process .4 on DFD 2, Figure B-3)**Input/Output:**

Packet_header : data_in
 Packet_data_field : data_in
 Eng_Mem_data : data_out
 Unpacked_radiance_packet_data : data_out
 Current_HK_telem : data_out
 Prior_HK_telem : data_out
 Sci_eng_data : data_out

Requirements:

- PR01-F-2.4-1. Shall unpack all radiance data from 12-bits in the MODIS_pkt to Unpacked_MODIS_radiance when the packet contains radiance data, using the format documented in SBRS CDRL 305.
- PR01-F-2.4-2. Shall unpack all engineering or memory data from the MODIS_pkt to Eng_Mem_data when the packet contains engineering or memory data, using the format documented in SBRS CDRL 305.
- PR01-F-2.4-3. Deleted
- PR01-F-2.4-4. Deleted

Requirements:

PR01-F-2.5-1. Deleted
PR01-F-2.5-2. Deleted
PR01-F-2.5-3. Deleted
PR01-F-2.5-4. Deleted
PR01-F-2.5-5. Deleted
PR01-F-2.5-6. Deleted

PR01-2.6 Write_failed_packet (Process .6 on DFD 2, Figure B-3)**Input/Output:**

MODIS_pkt : data_in
Failed_pkts : data_out

Requirement:

PR01-F-2.6. Shall write MODIS_pkt that has failed any validation check to Failed_pkts.

PR01-2.7 Extract_MODIS_pkt_info (Process .7 on DFD 2.2, Figure B-3)**Input/Output:**

MODIS_pkt : data_in
Packet_data_field : data_out
MODIS_pkt : data_out

Requirements:

PR01-F-2.7-1. Shall extract all fields from the CCSDS primary header, the CCSDS secondary header and the MODIS header, using the format documented in SBRS CDRL 305.
PR01-F-2.7-2. Shall extract the Packet_data_field from within the MODIS-pkt, using the format documented in SBRS CDRL 305.
PR01-F-2.7-3. Shall extract the Packet_checksum from within the MODIS-pkt, using the format documented in SBRS CDRL 305.

PR01-3 Create_MODIS_Scans (Process 3 on DFD 0, Figure B-1)**Input/Output:**

Decommutated_eng_mem_list : data_in
Eng_Mem_data : data_in
Packet_header : data_in
Unpacked_radiance_packet_data : data_in
Scan_log_msgs : data_out
MODIS_scan : data_out

See DFD 3 (Figure B-5) for decomposition.

PR01-3.1 Accumulate_radiance_data (Process .1 on DFD 3, Figure B-5)**Input/Output:**

Unpacked_radiance_packet_data : data_in

MODIS_scan_radiances : data_out

MODIS_scan_level_metadata : data_out

Requirements:

PR01-F-3.1-1. Shall place the Unpacked_radiance_packet_data into the appropriate location in MODIS_scan as MODIS_scan_radiances.

PR01-F-3.1-2. Shall accumulate MODIS_scan_level_metadata.

PR01-F-3.1-3. Deleted

PR01-3.2 Write_scan_to_output_product (Process .2 on DFD 3, Figure B-5)**Input/Output:**

MODIS_scan_data : data_in

Scan_log_msgs : data_out

MODIS_scan : data_out

Requirements:

PR01-F-3.2-1. Shall calculate the MODIS_scan_level_metadata for the each MODIS_scan.

PR01-F-3.2-2. Shall output each MODIS_scan to the current Level1A_data product.

PR01-F-3.2-3. Deleted

PR01-F-3.2-4. Shall write Scan_log_msgs to Log_messages when errors occur in the Write_scan_to_output_product process.

PR01-3.3 Accumulate_eng_mem_data (Process .3 on DFD 3, Figure B-5)**Input/Output:**

Decommutated_eng_mem_list : data_in

Eng_Mem_data : data_in

Scan_eng_mem_data : data_out

Requirements:

PR01-F-3.3-1. Shall store Eng_Mem_data in the current MODIS_scan.

PR01-F-3.3-2. Shall store Decommutated_eng_mem_list in the current MODIS_scan.

PR01-F-3.3-3. Deleted

PR01-4 Create_Level_1A_Products (Process 4 on DFD 0, Figure B-1)**Input/Output:**

Failed_pkts : data_in

MODIS_scan : data_in

Init_metadata : data_in

Level1A_data : data_out

Product_log_msgs : data_out

Requirements:

- PR01-F-4-1. Shall create each Level1A_data product.
- PR01-F-4-2. Shall accumulate one or more occurrences of MODIS_scan to populate each Level1A_data product.
- PR01-F-4-3. Shall create ECS_standard_global_metadata and MODISL1A_specific_global_metadata for each completed Level1A_data product, using Level0_open_metadata and selected metadata contained in MODIS_scan.
- PR01-F-4-4. Shall close each Level1A_data product.
- PR01-F-4-5. Shall write Product_log_msgs to Log_messages when errors occur in the Create_Level_1A_product.
- PR01-F-4-6. Deleted
- PR01-F-4-7. Shall write Failed_pkts (if any) to Level 1A_data product.

PR01-5 Create_decommutated_eng_mem_list (Process 5 on DFD 0, Figure B-1)**Input/Output:**

Current_HK_telem : data_in
 Prior_HK_telem : data_in
 Decommuted_eng_mem_list : data_out

See DFD 5 (Figure B-6) for decomposition.

PR01-5.1 Unpack_HK_telem_data (Process .1 on DFD 5, Figure B-6)**Input/Output:**

Current_HK_telem : data_in
 Prior_HK_telem : data_in
 Sci_eng_data : data_in
 Decommuted_eng_mem_list : data_out

Requirements:

- PR01-F-5.1-1. Shall unpack the information contained in Current_HK_telem, Prior_HK_telem, and Sci_eng_data, using the format described in SBRS CDRL 305.
- PR01-F-5.1-2. Shall update Decommuted_eng_mem_list with the values extracted from Current_HK_telem, Prior_HK_telem, and Sci_eng_data,
- PR01-F-5.2-1. Deleted
- PR01-F-5.2-2. Deleted

PR01-6 Close_processing_run (Process .6 on DFD 0, Figure B-1)**Input/Output:**

Level0_data : data_in
 LevelA_data : data_in
 Close_log_msgs : data_out

Requirements:

- PR01-F-6.1. Shall close all Level 0 files upon completion of L1A granules.
- PR01-F-6.2. Shall close all Level 0 files upon receipt of Program_stop.
- PR01-F-6.3. Shall write Close_log_msgs to Log_messages when errors occur in the Close_processing_run process.

5.1.2 Geolocation Process and Data Requirements

PR03-1 Read Parameters (Process 1 on DFD 0, Figure D-1)

Input/Output:

Geolocation_Parameters : data_in
 process_control_parameters: data_in
 instrument_parameters : data_out
 instrument_timing: data_out
 control_point_parameters : data_out
 level1a_conversion_and_validation_parameters : data_out
 geolocation_parameter_metadata : data_out
 focal_plane_geometry : data_out
 Ephemeris_source: data_out
 parameter_input_status_messages : data_out

Requirements:

- PR03-F-1-1. Shall read the Geolocation_Parameters.
 PR03-F-1-2. Shall report errors as parameter_input_status_messages in log status message file through the use of SDP functions.
 PR03-F-1-3. Shall read the process_control_parameters from the SDP process control file as: Ephemeris_source.

PR03-2 Process Level 1A Data (Process 2 on DFD 0, Figure D-1)

Input/Output:

Level_1A_Data : data_in
 level1a_conversion_and_validation_parameters : data_in
 instrument_timing: data_in
 scan_quality_flags : data_out
 validated_level_1a_data : data_out
 level1a_global_metadata : data_out
 granule_level_state : data_out
 data_input_status_messages : data_out

See DFD 2 (Figure D-2) for decomposition.

PR03-2.1 Read Level 1A Data (Process .1 on DFD 2, Figure D-2)

Input/Output:

level1a_conversion_and_validation_parameters : data_in
 Level_1A_Data : data_in
 data_input_status_messages : data_out
 level1a_global_metadata : data_out
 spacecraft_ancillary_data : data_out
 level1a_scan_flags : data_out
 science_abnormal: data_out
 science_state: data_out
 instrument_scan_data : data_out

Requirements:

- PR03-F-2.1-1. Shall read the Level_1A_Data as level1a_global_metadata, level1a_scan_flags, spacecraft_ancillary_data, and instrument_scan_data.
- PR03-F-2.1-2. Shall generate data_input_status_messages if errors are encountered in reading Level_1A_Data.

PR03-2.2 Convert orbit and attitude data (Process .2 on DFD 2, Figure D-2)**Input/Output:**

spacecraft_ancillary_data_conversion_parameters : data_in
 spacecraft_ancillary_data : data_in
 converted_spacecraft_ancillary_data : data_out

Requirement:

- PR03-F-2.2-1. Shall convert spacecraft_ancillary_data to converted_spacecraft_ancillary_data according to the GISS, using spacecraft_ancillary_data_conversion_parameters.

PR03-2.3 Convert mirror encoder data (Process .3 on DFD 2, Figure D-2)**Input/Output:**

instrument_scan_data : data_in
 instrument_data_conversion_parameters : data_in
 converted_instrument_scan_data : data_out

Requirement:

- PR03-F-2.3-1. Shall convert instrument_scan_data to converted_instrument_scan_data according to the specifications in CDRL 305, using the instrument_data_conversion_parameters.

PR03-2.4 Detect corrupt orbit and attitude data values (Process .4 on DFD 2, Figure D-2)**Input/Output:**

converted_spacecraft_ancillary_data : data_in
 science_abnormal: data_in
 spacecraft_ancillary_data_validation_parameters: data_in
 data_input_status_messages : data_out
 validated_spacecraft_ancillary_data : data_out

Requirements:

- PR03-F-2.4-1. Shall detect corrupt converted_spacecraft_ancillary_data to produce validated_spacecraft_ancillary_data.
- PR03-F-2.4-2. Shall generate data_input_status_messages when corrupted data values are detected.

PR03-2.5 Produce mirror time-position sample (Process .5 on DFD 2, Figure D-2)**Input/Output:**

level1a_scan_flags : data_in
 converted_instrument_scan_data : data_in
 encoder_time: data_in
 sample_impulse: data_in
 instrument_data_validation_parameters: data_in
 data_input_status_messages : data_out
 scan_quality_flags : data_out
 validated_instrument_data : data_out

Requirements:

- PR03-F-2.5-1. Shall detect corrupt
 converted_instrument_scan_data.converted_mirror_encoder_data.
- PR03-F-2.5-2. Shall generate data_input_status_messages and set
 scan_quality_flags if corrupted data values are detected.
- PR03-F-2.5-3. Shall produce
 validated_mirror_encoder_data.mirror_encoder_times from
 converted_instrument_scan_data.converted_mirror_encoder_data
 as described in the ATBD, section 3.1.4.1.
- PR03-F-2.5-4. Shall produce
 validated_mirror_encoder_data.mirror_encoder_positions from
 converted_instrument_scan_data.scan_mirror_start_positions
 as described in the ATBD, section 3.1.4.1.
- PR03-F-2.5-5. Shall incorporate level1a_scan_flags into scan_quality flags.

PR03-2.6 Identify abnormal granule state (Process .6 on DFD 2, Figure D-2)**Input/Output:**

science_state : data_in
 science_abnormal : data_in
 granule_level_state : data_out

Requirements:

- PR03-F-2.6-1. Shall set the respective granule_science_state or
 granule_science_abnormal flags if either science_state or
 science_abnormal scan-level flags indicate that abnormal
 instrument or spacecraft state occurred.

PR03-3 Compute Geolocation Fields (Process 3 on DFD 0, Figure D-1)**Input/Output:**

instrument_parameters : data_in
 geolocation_scan_data : data_out
 scan_quality_flags : data_in
 validated_level_1a_data : data_in
 Elevation_Data : data_in
 Spacecraft_Data : data_in

Ephemeris_source: data_in
computation_status_messages : data_out

See DFD 3 (Figure D-3) for decomposition.

PR03-3.1 Determine instrument lines of sight (Process .1 on DFD 3, Figure D-3)

Input/Output:

scan_quality_flags : data_in
instrument_parameters : data_in
validated_instrument_data : data_in
lines_of_sight : data_out
time_of_observation : data_out
spatial_element_quality_flags : data_out

See DFD 3.1 (Figure D-4) for decomposition.

PR03-3.1.1 Compute time of observation (Process .1 on DFD 3.1, Figure D-4)

Input/Output:

scan_quality_flags : data_in
scan_start_times : data_in
number_of_Earth_view_frames : data_in
instrument_timing : data_in
time_of_observation : data_out
spatial_element_quality_flags : data_out

Requirements:

- PR03-F-3.1.1-1. Shall compute the time_of_observation for each frame from the instrument_timing, number_of_Earth_view_frames, and scan_start_times as defined in the Geolocation ATBD, section 3.1.4.1.
- PR03-F-3.1.1-2. Shall set values of spatial_element_quality_flags to indicate insufficient or invalid input data.

PR03-3.1.2 Compute mirror scan angle (Process .2 on DFD 3.1, Figure D-4)

Input/Output:

time_of_observation : data_in
spatial_element_quality_flags : data_in
scan_quality_flags : data_in
mirror_sides : data_in
validated_mirror_encoder_data : data_in
mirror_model : data_in
spatial_element_quality_flags : data_out
mirror_scan_angle : data_out

Requirement:

- PR03-F-3.1.2-1. Shall compute the `mirror_scan_angle` as defined in the Geolocation ATBD, section 3.1.4.1, from the `time_of_observation`, `mirror_sides`, `mirror_model`, and `validated_mirror_encoder_data` interpolated using a Chebyshev polynomial fit.
- PR03-F-3.1.2-2. Shall set `spatial_element_quality_flags` to indicate insufficient or invalid input data.

PR03-3.1.3 Compute mirror normal vector (Process .3 on DFD 3.1, Figure D-4)**Input/Output:**

`mirror_scan_angle` : data_in
`mirror_model` : data_in
`mirror_normal_vector` : data_out

Requirement:

- PR03-F-3.1.3-1. Shall compute the `mirror_normal_vector` from the `mirror_model`, `mirror_scan_angle`, and `mirror_sides` as defined in the Geolocation ATBD, section 3.1.4.1.

PR03-3.1.4 Compute lines of sight (Process .3 on DFD 3.1, Figure D-4)**Input/Output:**

`mirror_normal_vector` : data_in
`focal_plane_geometry` : data_in
`internal_coordinate_transformations` : data_in
`lines_of_sight` : data_out

Requirement:

- PR03-F-3.1.4-1. Shall compute `lines_of_sight` from `mirror_normal_vector`, `focal_plane_geometry`, and `internal_coordinate_transformations` as defined in the Geolocation ATBD, section 3.1.4.1.

PR03-3.2 Determine observed locations (Process .2 on DFD 3, Figure D-3)**Input/Output:**

`Ephemeris_source` : data_in
`interpolated_spacecraft_data` : data_out
`Spacecraft_orbit_velocity_ECR` : data_out
`Spacecraft_orbit_position_ECR` : data_out
`validated_spacecraft_ancillary_data` : data_in
`lines_of_sight` : data_in
`time_of_observation` : data_in
`spatial_element_quality_flags` : data_in
`Spacecraft_Data` : data_in
`computation_status_messages` : data_out
`lines_of_sight_ECR` : data_out
`observed_locations` : data_out
`spatial-element_quality_flags` : data_out

See DFD 3.2 (Figure D-5) for decomposition.

PR03-3.2.1 Interpolate orbit and attitude data to observation time (Process .1 on DFD 3.2, Figure D-5)**Input/Output:**

spatial_element_quality_flags : data_in
validated_spacecraft_ancillary_data : data_in
time_of_observation : data_in
Spacecraft_Data : data_in
Ephemeris_source: data_in
computation_status_messages : data_out
interpolated_spacecraft_data : data_out
spatial_element_quality_flags : data_out

Requirements:

- PR03-F-3.2.1-1. Shall interpolate validated_spacecraft_ancillary_data or Spacecraft_Data to the time_of_observation to determine interpolated_spacecraft_data as defined in the Geolocation ATBD, section 3.1.4.2.
- PR03-F-3.2.1-2. Shall generate computation_status_messages and set spatial_element_quality_flags in response to interpolation errors.

PR03-3.2.2 Transform to ECR coordinates (Process .2 on DFD 3.2, Figure D-5)**Input/Output:**

interpolated_spacecraft_data : data_in
time_of_observation : data_in
lines_of_sight : data_in
spacecraft_orbit_position_ECR : data_out
spacecraft_orbit_velocity_ECR : data_out
spatial_element_quality_flags : data_in
lines_of_sight_ECR : data_out
spatial_element_quality_flags : data_out

Requirement:

- PR03-F-3.2.2-1. Shall transform the interpolated_spacecraft_data, and the lines_of_sight to the Earth-Centered Rotating (ECR) reference frame as defined in the Geolocation ATBD, section 3.1.4.2.

PR03-3.2.3 Determine intersection of line of sight with reference ellipsoid (Process .3 on DFD 3.2, Figure D-5)**Input/Output:**

spacecraft_orbit_position_ECR : data_in
lines_of_sight_ECR : data_in
spatial_element_quality_flags : data_in
computation_status_messages : data_out
ellipsoid_intersection : data_out
spatial_element_quality_flags : data_out

Requirements:

- PR03-F-3.2.3-1. Shall determine `ellipsoid_intersection` using `lines_of_sight` and `spacecraft_orbit_position_ECR` as defined in the Geolocation ATBD, section 3.1.4.2.
- PR03-F-3.2.3-2. Shall generate `computation_status_messages` and set `spatial_element_quality_flags` in response to errors in determining `ellipsoid_intersection`.

PR03-3.2.4 Compute geodetic latitude and longitude (Process .4 on DFD 3.2, Figure D-5)**Input/Output:**

`ellipsoid_intersection` : data_in
`spatial_element_quality_flags` : data_in
`geodetic_latitude` : data_out
`geodetic_longitude` : data_out
`spatial_element_quality_flags` : data_out

Requirement:

- PR03-F-3.2.4-1. Shall compute the `geodetic_latitude` and `geodetic_longitude` from the `ellipsoid_intersection` as defined in the Geolocation ATBD, section 3.1.4.2.

PR03-3.3 Correct observed locations (Process .3 on DFD 3, Figure D-3)**Input/Output:**

`lines_of_sight_ECR` : data_in
`observed_locations` : data_in
`spatial_element_quality_flags` : data_in
`Elevation_Data` : data_in
`spatial_element_quality_flags` : data_out
`observed_locations_with_terrain_correction` : data_out

See DFD 3.3 (Figure D-6) for decomposition.

PR03-3.3.1 Compute intersection of line of sight with terrain surface (Process .1 on DFD 3.3, Figure D-6)**Input/Output:**

`terrain_height` : data_in
`observed_locations` : data_in
`lines_of_sight_ECR` : data_in
`terrain_intersection` : data_out

Requirement:

- PR03-F-3.3.1-1. Shall compute the `terrain_intersection` of the `lines_of_sight` ECR with the `terrain_height` for the `observed_locations` (`geodetic_latitude` and `geodetic_longitude`) as defined in the Geolocation ATBD, section 3.1.4.2.

PR03-3.3.2 Compute height geodetic latitude and longitude (Process .2 on DFD 3.3, Figure D-6)**Input/Output:**

terrain_intersection : data_in
observed_locations_with_terrain_correction : data_out

Requirement:

PR03-F-3.3.2-1. Shall compute the observed_locations_with_terrain_correction from the terrain_intersection as defined in the Geolocation ATBD, section 3.1.4.2.

PR03-3.3.3 Read DEM data for observed locations (Process .3 on DFD 3.3, Figure D-6)**Input/Output:**

Elevation_Data : data_in
observed_locations : data_in
spatial_element_quality_flags : data_out
spatial_element_quality_flags : data_in
terrain_height : data_out

Requirements:

PR03-F-3.3.3-1. Shall read Elevation_Data.digital_elevation_model_data to determine terrain_height for region surrounding observed_locations.
PR03-F-3.3.3-2. Shall set spatial_element_quality_flags to indicate Elevation_Data of inferior quality.

PR03-3.4 Compute sensor and solar vectors (Process .4 on DFD 3, Figure D-3)**Input/Output:**

observed_locations_with_terrain_correction : data_in
spacecraft_orbit_position_ECR : data_in
time_of_observation : data_in
sun_reference_vector : data_out
sensor_and_solar_vectors : data_out

See DFD 3.4 (Figure D-7) for decomposition.

PR03-3.4.1 Determine vector to spacecraft position (Process .1 on DFD 3.4, Figure D-7)**Input/Output:**

spacecraft_orbit_position_ECR : data_in
observed_locations_with_terrain_correction : data_in
vector_from_observed_locations_to_spacecraft_position : data_out

Requirement:

- PR03-F-3.4.1-1. Shall determine the `vector_from_observed_locations_to_spacecraft_position` from the `observed_locations_with_terrain_correction` as defined in the Geolocation ATBD, section 3.1.4.2.

PR03-3.4.2 Compute range, sensor zenith and azimuth angles (Process .2 on DFD 3.4, Figure D-7)**Input/Output:**

`spatial_element_quality_flags` : data:out
`spatial_element_quality_flags` : data:in
`vector_from_observed_locations_to_spacecraft_position` : data_in
`sensor_vectors` : data_out
`observed_locations_with_terrain_correction` : data_in

Requirement:

- PR03-F-3.4.2-1. Shall compute the `sensor_vectors` for the `observed_locations_with_terrain_corrections` from the `vector_from_observed_location_to_spacecraft_position` as defined in the Geolocation ATBD, section 3.1.4.2.
- PR03-F-3.4.2-2. Shall set `spatial_element_quality_flags` to identify unavailable or oversize sensor range.

PR03-3.4.3 Determine Earth to Sun vector (ECR) (Process .3 on DFD 3.4, Figure D-7)**Input/Output:**

`time_of_observation` : data_in
`sun_reference_vector` : data_out

Requirement:

- PR03-F-3.4.3-1. Shall determine the `sun_reference_vector` in the ECR reference frame at the `time_of_observation`.

PR03-3.4.4 Compute solar zenith and azimuth angles (Process .4 on DFD 3.4, Figure D-7)**Input/Output:**

`sun_reference_vector` : data_in
`observed_locations_with_terrain_correction` : data_in
`solar_vectors` : data_out

Requirement:

- PR03-F-3.4.4-1. Shall compute `solar_vectors` at the observed location from the `sun_reference_vector` and the `observed_locations_with_terrain_correction` as defined in the Geolocation ATBD, section 3.1.4.2.

PR03-3.5 Compute solar diffuser angles (Process .5 on DFD 3, Figure D-3)**Input/Output:**

scan_quality_flags : data_in
validated_spacecraft_ancillary_data : data_in
solar_diffuser_view_data : data_in
internal_coordinate_transformations : data_in
Spacecraft_Data : data_in
solar_diffuser_angles : data_out

See DFD 3.5 (Figure B-7) for decomposition.

PR03-3.5.1 Interpolate orbit and attitude data to solar diffuser time (Process .1 on DFD 3.5, Figure D-7)**Input/Output:**

Spacecraft_Data : data_in
scan_quality_flags : data_in
validated_spacecraft_ancillary_data : data_in
solar_diffuser_start_times : data_in
number_of_solar_diffuser_frames : data_in
interpolated_spacecraft_data : data_out

Requirement:

PR03-F-3.5.1-1. Shall interpolate validated_spacecraft_ancillary_data or Spacecraft_Data using the solar_diffuser_start_times and number_of_solar_diffuser_frames to determine interpolated_spacecraft_data.

PR03-3.5.2 Determine Earth to Sun Vector (Process .2 on DFD 3.5, Figure D-7)**Input/Output:**

solar_diffuser_start_times : data_in
scan_quality_flags : data_in
sun_reference_vector : data_out
number_of_solar_diffuser_frames : data_in

Requirement:

PR03-F-3.5.2-1. Shall determine the sun_reference_vector in the ECR reference frame at the mid-point of the solar diffuser view sector using the solar_diffuser_start_times and the number_of_solar_diffuser_frames.

PR03-3.5.3 Determine solar diffuser angles (Process .3 on DFD 3.5, Figure D-7)**Input/Output:**

interpolated_spacecraft_data : data_in
internal_coordinate_transformations : data_in
sun_reference_vector : data_in
solar_diffuser_angles : data_out

Requirements:

- PR03-F-3.5.3-1. Shall determine the ECR reference frame-to-solar diffuser coordinate system transformation using the `interpolated_spacecraft_data` and the `internal_coordinate_transformations`.
- PR03-F-3.5.3-2. Shall transform the `sun_reference_vector` to the solar diffuser coordinate system.
- PR03-F-3.5.3-3. Shall compute the `solar_diffuser_angles` from the transformed Sun vector.

PR03-3.6 Compute lunar vectors (Process .6 on DFD 3, Figure D-3)**Input/Output:**

`scan_quality_flags` : data_in
`validated_spacecraft_ancillary_data` : data_in
`space_view_data` : data_in
`internal_coordinate_transformations` : data_in
`Spacecraft_Data` : data_in
`lunar_vector` : data_out

See DFD 3.6 (Figure D-9) for decomposition.

PR03-3.6.1 Interpolate orbit and attitude data to space_view time (Process .1 on DFD 3.6, Figure D-9)**Input/Output:**

`Spacecraft_Data` : data_in
`scan_quality_flags` : data_in
`validated_spacecraft_ancillary_data` : data_in
`space_view_start_times` : data_in
`number_of_space_view_frames` : data_in
`interpolated_spacecraft_data` : data_out

Requirement:

- PR03-F-3.6.1-1. Shall interpolate `validated_spacecraft_ancillary_data` or `Spacecraft_Data` using the `space_view_start_times` and `number_of_space_view_frames` to determine `interpolated_spacecraft_data`.

PR03-3.6.2 Determine Earth to Moon Vector (Process .2 on DFD 3.6, Figure D-9)**Input/Output:**

`space_view_start_times` : data_in
`scan_quality_flags` : data_in
`moon_reference_vector` : data_out
`number_of_space_view_frames` : data_in

Requirement:

PR03-F-3.6.2-1. Shall determine the moon_reference_vector in the ECR reference frame at the mid-point of the space view sector using the space_view_start_times and the number_of_space_view_frames.

PR03-3.6.3 Determine lunar vectors (Process .3 on DFD 3.6, Figure D-9)**Input/Output:**

interpolated_spacecraft_data : data_in
internal_coordinate_transformations : data_in
moon_reference_vector : data_in
lunar_vector : data_out

Requirements:

PR03-F-3.6.3-1. Shall determine the ECR reference frame-to-instrument coordinate system transformation using the interpolated_spacecraft_data and the internal_coordinate_transformations.

PR03-F-3.6.3-2. Shall transform the moon_reference_vector to the instrument coordinate system to determine the lunar_vector.

PR03-3.7 Determine Earth View center Ephemeris (Process .7 on DFD 3, Figure D-3)**Input/Output:**

interpolated_spacecraft_data : data_in
time_of_observation : data_in
number_of_Earth_view_frames : data_in
spacecraft_orbit_position_ECR : data_in
spacecraft_orbit_velocity_ECR : data_in
sun_reference_vector : data_in
Earth_View_Center_data : data_out

Requirement:

PR03-F-3.7-1. Shall determine Earth_View_Center_data as the spacecraft ephemeris and the sun reference vector at the Earth View's (odd) center frame's center time, as determined from number_of_Earth_view_frames, time_of_observation, interpolated_spacecraft_data, spacecraft_orbit_position_ECR, spacecraft_orbit_velocity_ECR, and sun_reference_vector.

PR03-3.8 Determine Land/Sea mask (Process .8 on DFD 3, Figure D-3)**Input/Output:**

observed_locations_with_terrain_correction : data_in
land-sea_mask : data_in
land-sea_mask_at_observed_locations: data_out

Requirement:

PR03-F-3.8-1. Shall determine the land-sea_mask_at_observed_locations at the observed_locations_with_terrain_correction from the land-sea_mask data.

PR03-4 Write Geolocation Product (Process 4 on DFD 0, Figure D-1)**Input/Output:**

geolocation_scan_data : data_in
level1a_global_metadata : data_in
focal_plane_geometry : data_in
geolocation_parameter_metadata : data_in
Ephemeris_source: data_in
data_output_status_messages : data_out
Geolocation_Data : data_out

See DFD 4 (Figure D-10) for decomposition.

PR03-4.1 Write Geolocation Global Metadata (Process .1 on DFD 4, Figure D-10)**Input/Output:**

level1a_global_metadata : data_in
granule_level_state: data_in
geolocation_scan_data : data_in
Ephemeris_source: data_in
data_output_status_messages : data_out
geolocation_global_metadata : data_out

Requirements:

- PR03-F-4.1-1. Shall write the Geolocation Global Metadata deduced from the geolocation_scan_data, granule_level_state, and the level_1A_global_metadata as: geolocation_global_metadata.
- PR02-F-4.1-2. Shall generate data_output_status_messages if write errors are encountered.

PR03-4.2 Write Instrument Parameters (Process .2 on DFD 4, Figure D-10)**Input/Output:**

focal_plane_geometry : data_in
geolocation_parameter_metadata : data_in
data_output_status_messages : data_out
focal_plane_geometry : data_out
geolocation_parameter_metadata : data_out

Requirements:

- PR03-F-4.2-1. Shall write the focal_plane_geometry and geolocation_parameter_metadata to the Geolocation Product.
- PR03-F-4.2-2. Shall generate data_output_status_messages if write errors are encountered.

PR03-4.3 Write Geolocation Scan Data (Process .3 on DFD 4, Figure D-10)**Input/Output:**

geolocation_scan_data : data_in
data_output_status_messages : data_out
product_fields_per_spatial_element : data_out
product_fields_per_scan : data_out

Requirements:

- PR03-F-4.3-1. Shall write the geolocation_scan_data as:
product_fields_per_spatial_element and product fields_per_scan.
- PR03-F-4.3-2. Shall generate data_output_status_messages if write errors are encountered.
- PR03-F-4.3-3. Shall write the product_fields_per_spatial_element in HDF-EOS Swath format.

PR03-4.4 Write Spatial Metadata to L1A Product(Process .4 on DFD 4, Figure D-10)**Input/Output:**

geolocation_scan_data : data_in
data_output_status_messages : data_out
spatial_metadata : data_out

Requirements:

- PR03-F-4.4-1. Shall write the Spatial Metadata deduced from the geolocation_scan_data and the level_1A_global_metadata to the L1A Product as: spatial_metadata.
- PR02-F-4.4-2. Shall generate data_output_status_messages if write errors are encountered.

PR03-5 Assess Geolocation Accuracy (Process 5 on DFD 0, Figure D-1)**Input/Output:**

Control_Points : data_in
control_point_parameters : data_in
Level_1B_Data : data_in
Level_2_Data : data_in
geolocation_scan_data : data_in
Control_Point_Matchups : data_out

See DFD 5 (Figure D-11) for decomposition.

PR03-5.1 Control Point Selection (Process .1 on DFD 5, Figure D-11)**Input/Output:**

Control_Points : data_in
geolocation_scan_data : data_in
control_point_exclusion_mask : data_in
control_point_parameters : data_in
selected_ocean_control_points : data_out
selected_land_control_points : data_out

Requirements:

- PR03-F-5.1-1. Shall read `land_control_points` and generate `selected_land_control_points` based on geographic boundaries represented by `geolocation_scan_data` and the `control_point_exclusion_mask`.
- PR03-F-5.1-2. Shall read `ocean_control_points` and generate `selected_ocean_control_points` based on geographic boundaries represented by `geolocation_scan_data` and the `control_point_exclusion_mask`.

PR03-5.2 Land Control Point Correlation (Process .2 on DFD 5, Figure D-11)**Input/Output:**

`geolocation_scan_data` : data_in
`land_control_point_parameters` : data_in
`land_control_point_radiances` : data_in
`selected_land_control_points` : data_in
`land_control_point_matchups` : data_out

Requirements:

- PR03-F-5.2-1. Shall perform correlation of `selected_land_control_points` with `land_control_point_radiances` and `geolocation_scan_data` to generate `land_control_point_matchups` as specified in the Geolocation ATBD.
- PR03-F-5.2-2 Shall write land control point residuals data as:
`land_control_point_matchups`.

PR03-5.3 Ocean Control Point Correlation (Process .3 on DFD 5, Figure D-11)**Input/Output:**

`geolocation_scan_data` : data_in
`selected_ocean_control_points` : data_in
`ocean_control_point_radiances` : data_in
`ocean_control_point_parameters` : data_in
`ocean_control_point_matchups` : data_out

Requirement:

- PR03-F-5.3-1. Shall perform correlation of `selected_ocean_control_points` with `ocean_control_point_radiances` and `geolocation_scan_data` to generate `ocean_control_point_matchups` as specified in the Geolocation ATBD.
- PR03-F-5.3-2. Shall write ocean control point residuals data as:
`ocean_control_point_matchups`.

PR03-5.4 Read Level 1B Data (Process .4 on DFD 5, Figure D-11)**Input/Output:**

Level_1B_Data : data_in
 data_input_status_messages : data_out
 control_point_radiances : data_out
 level1B_global_metadata : data_out
 level1B_scan_flags : data_out

Requirements:

- PR03-F-5.4-1. Shall read the Level_1B_Data as level1B_global_metadata, level1B_scan_flags, and control_point_radiances.
 PR03-F-5.4-2. Shall generate data_input_status_messages if errors are encountered in reading Level_1B_Data.

PR03-5.5 Read Level 2 Data (Process .5 on DFD 5, Figure D-11)**Input/Output:**

Level_2_Data : data_in
 data_input_status_messages : data_out
 level_2_Classification_masks : data_out
 level2_global_metadata : data_out
 level2_scan_flags : data_out

Requirements:

- PR03-F-5.5-1. Shall read the Level_2_Data for the MODIS Classification Masks (MOD35) and other MODIS Level 2 products that may contribute to the identification of regions in a granule that should be excluded for the purpose of geolocation process accuracy assessment as level2_global_metadata, level2_scan_flags, and L2_classification_masks.
 PR03-F-5.5-2. Shall generate data_input_status_messages if errors are encountered in reading Level_2_Data.

PR03-5.6 Identify regions usable for Control Point Correlation (Process .6 on DFD 5, Figure D-11)**Input/Output:**

scan_quality_flags : data_in
 spatial_element_quality_flags: data_in
 level1B_scan_flags: data_in
 level1B_global_metadata : data_in
 level_2_Classification_masks : data_in
 level2_global_metadata : data_in
 level2_scan_flags: data_in
 control_point_exclusion_mask : data_out

Requirement:

PR03-F-5.6-1. Shall determine the control_point_exclusion_mask of regions not to be used for control point correlation because of bad validation or quality of the data or because of cloud coverage or other scene distortion from the geolocation scan_quality_flags and spatial_element_quality_flags, level1B_scan_flags, level1B_global_metadata, level_2_Classification_masks, level2_global_metadata, and level2_scan_flags.

PR03-5.7 Write Control Point Metadata (Process .7 on DFD 5, Figure D-11)**Input/Output:**

geolocation_scan_data : data_in
control_point_parameters : data_in
control_point_metadata : data_out

Requirements:

PR03-F-5.7-1. Shall write the control_point_metadata derived from control_point_parameters and geolocation_scan_data to each control point residuals file.

PR03-6 Write Geolocation Initiation and Termination messages (Process 6 on DFD 0, Figure D-1)**Input/Output:**

initiation_messages : data_out
termination_messages : data_out

Requirements:

PR03-F-6.1-1. The L1A/Geolocation software shall generate initiation_messages at program commencement.
PR02-F-6.2-1. The L1A/Geolocation software shall generate termination_messages at program completion.

5.2 Performance and Quality Engineering Requirements

There are no performance and quality engineering requirements for either the L1A or Geolocation software.

5.3 Safety Requirements

There are no safety requirements for either the L1A or Geolocation software.

5.4 Security and Privacy Requirements

There are no security or privacy requirements for either the L1A or Geolocation software.

5.5 Implementation Constraints

- PR01-I-1, PR03-I-1 The L1A/Geolocation software shall follow the coding standards established by the MODIS project in MODIS Software Development Standards and Guidelines.
- PR01-I-2, PR03-I-2 The L1A/Geolocation software shall use all applicable mandatory SDPTK 5.2 routines and any additional SDPTK routines that are useful.
- PR01-I-3, PR03-I-3 The L1A/Geolocation software shall use HDF Version 4.1r1 for reading and writing all HDF files.
- PR01-I-4, PR03-I-4 The L1A/Geolocation software shall use HDF-EOS Version 2.0.
- PR01-I-5, PR03-I-5 The L1A/Geolocation software processes shall be able to process input data sets with the following nominal attributes:
- Time transitions, including Greenwich Meridian crossing, leap year, beginning of the year, beginning of the millennium, and/or start of data collection not on a granule boundary.
 - Terminator crossing , International Date Line crossing, and/or Earth pole within a granule.
 - Day, Night, or mixed mode data.
- PR01-I-6, PR03-I-6 The L1A/Geolocation software processes shall handle input data sets with the following error attributes:
- Fill data, including filled pixels, filled bands, and/or filled scans.
 - Redundant data, consisting of repeated packets at Level 0 or duplicate input files at all product levels.
 - Corrupted data as indicated by Quality Assurance (QA) flags also located within the file.
 - Missing input MODIS product files, ancillary files, and look-up tables.
 - Wrong format input files.
 - Noisy or dead detectors, as indicated by QA flags located within the Level 1B file.

5.6 Site Adaptation

- PR01-S-1, PR03-S-1 The L1A and Geolocation software shall provide interfaces to mandatory SDPTK elements such as the Status Message Facility (SMF), the Process Control File (PCF), and the Metadata Configuration File (MCF).

5.7 Design Goals

- 5.7-1 The L1A and Geolocation software for V1 should reuse as much of the Beta version's design and code as possible.
- 5.7-2 L1A and Geolocation software design for V1 should be changed from the Beta design in response to several factors:
- new information about input data,
 - addition of new functionality to meet requirements,
 - improvement of software efficiency, and
 - improvement of software design.
- 5.7-3 The L1A and Geolocation software for V1 should be developed on an Silicon Graphics Inc. (SGI) platform, operating under IRIX 6.1 or higher operating system running in native (n32- or 64-bit) mode.

NOTE: These are not requirements.

(This page intentionally left blank)

6. REQUIREMENTS TRACEABILITY

The following table contains the traceability between the L1A and Geolocation requirements specified in this document, and parent requirements. The traceability column of the table contains the requirements number from the MODIS Science Data Processing Software Requirements Specification Version 2 and Beyond to which a given L1A or Geolocation requirements traces.

Table 6-1 L1A/Geolocation Requirements Traceability

Req. #	Requirement	Traceability
PR01-F-1.1-1	Shall open the Level0_data file.	5.2-3
PR01-F-1.1-2	Shall abort processing if the Level0_data file open fails.	5.2-2
PR01-F-1.1-3	Shall write Level0_open_metadata.	4.2-1
PR01-F-1.1-5	Shall write Level0_open_log_msgs to Log_messages when errors occur in the Open_Level0_file process.	5.2-1
PR01-F-1.2-1	Shall verify that the Spacecraft_indicator contained in the Level1A_Parameters is consistent with the Spacecraft_indicator found in the Level 0 file header.	5.2-4.5
PR01-F-1.2-2	Shall abort processing if Spacecraft_indicator verification fails.	5.2-2
PR01-F-1.2-5	Shall write Level0_header_log_msgs to Log_messages if the Spacecraft_indicator verification fails.	5.2-1
PR01-F-2.1-1	Shall read MODIS_packets from the Level0_data file.	5.2-3
PR01-F-2.1-3	Shall issue Program_stop if end-of-file condition is detected.	5.2-2
PR01-F-2.1-4	Shall write Read_packet_log_msgs to Log_messages when errors occur in the Read_MODIS_packet process.	5.2-1
PR01-F-2.1-5	Shall issue Program_stop if MODIS_packets cannot be read from Level0_data file.	5.2-1, 5.2-2
PR01-F-2.2.2-1	Shall verify that the values of the CCSDS primary header fields are consistent with values contained within SBRS CDRL 305.	5.3-1
PR01-F-2.2.2-2	Shall Abort_current_packet_processing if any CCSDS primary header field fails verification.	5.2-4.3
PR01-F-2.2.2-4	Shall write Verify_CCSDS_log_msgs to Log_messages when errors occur in the Verify_CCSDS_header_info process.	5.2-1
PR01-F-2.2.3-1	Shall verify that the values of all MODIS header fields are within ranges specified in SBRS CDRL 305.	5.3-1
PR01-F-2.2.3-2	Shall Abort_current_packet_processing if a MODIS packet fails verification of any MODIS header field.	5.2-4.3
PR01-F-2.2.3-4	Shall write Verify_MODIS_log_msgs to Log_messages when errors occur in the Verify_MODIS_header_info process.	5.2-1
PR01-F-2.2.4-1	Shall calculate the packet checksum using an algorithm to be supplied by SBRS.	5.3-1
PR01-F-2.2.4-2	Shall Abort_current_packet_processing if the calculated packet checksum is not the same as the packet_checksum in the MODIS_pkt.	5.2-4.3
PR01-F-2.2.4-4	Shall write Verify_checksum_log_msgs to Log_messages when errors occur in the Verify_packet_checksum process.	5.2-1
PR01-F-2.3-1	Shall determine the type of packet from information in the CCSDS secondary header and the MODIS header	5.2-3.3
PR01-F-2.3-2	Shall determine if the packet belongs in the current scan.	5.2-3
PR01-F-2.3-3	Shall remove redundant packets.	5.2-4.2
PR01-F-2.3-4	Shall determine if packets are missing from the current scan.	5.2-4.4
PR01-F-2.3-5	Shall write Missing_packet_log_msgs to Log_messages when missing packets are detected.	5.2-1

Req. #	Requirement	Traceability
PR01-F-2.4-1	Shall unpack all radiance data from 12-bits in the MODIS_pkt to unpacked_MODIS_radiance when the packet contains radiance data, using the format documented in SBR5 CDRL 305.	4.1-1
PR01-F-2.4-2	Shall unpack all engineering or memory data from the MODIS_pkt when to current_packet_state indicates that the data within the packet isEng_Mem_data when the packet contains engineering or memory data, using the format documented in SBR5 CDRL 305.	4.1-1
PR01-F-2.6	Shall write MODIS_pkt that has failed any validation check to Failed_pkts.	5.2-4.3
PR01-F-2.7-1	Shall extract all fields from the CCSDS primary header, the CCSDS secondary header and the MODIS header, using the format documented in SBR5 CDRL 305.	5.2-4.3
PR01-F-2.7-2.	Shall extract the Packet_data_field from within the MODIS-pkt, using the format documented in SBR5 CDRL 305.	5.2-4.3
PR01-F-2.7-3	Shall extract the Packet_checksum from within the MODIS-pkt, using the format documented in SBR5 CDRL 305.	5.2-4.3
PR01-F-3.1-1	Shall place the Unpacked_radiance_packet_data into the appropriate location in MODIS_scan as MODIS_scan_radiances.	4.1-1, 5.2-3.3
PR01-F-3.1-2	Shall accumulate MODIS_scan_level_metadata.	4.3-3
PR01-F-3.2-1	Shall calculate the MODIS_scan_level_metadata for each MODIS_scan.	4.3-3
PR01-F-3.2-2	Shall output each MODIS_scan to the current Level1A_data product.	4.1-2
PR01-F-3.2-4	Shall write Scan_log_msgs to Log_messages when errors occur in the Write_scan_to_output_product process.	3.4-1
PR01-F-3.3-1	Shall store Eng_Mem_data in the current MODIS_scan.	4.1-1
PR01-F-3.3-2	Shall store Decommuted_eng_mem_list in the current MODIS_scan.	4.1-1
PR01-F-4-1	Shall create each Level1A_data product.	4.1-1
PR01-F-4-2	Shall accumulate one or more occurrences of MODIS_scan to populate each Level1A_data product.	4.1-1
PR01-F-4-3	Shall create ECS_standard_global_metadata and MODISL1A_specific_global_metadata for each completed Level1A_data product, using Level0_open_metadata and selected metadata contained in the MODIS_scan.	4.2-1, 4.2-2, 4.2-3
PR01-F-4-4	Shall close each Level1A_data product.	5.1-3
PR01-F-4-5	Shall write Product_log_msgs to Log_messages when errors occur in the Create_Level_1A_product.	5.2-1
PR01-F-4-7	Shall write Failed_pkts (if any) to Level 1A_data product.	4.1-1
PR01-F-5.1-1	Shall unpack the information contained in Current_HK_telem, Prior_HK_telem, and Sci_eng_data, using the format described in SBR5 CDRL 305.	4.1-1
PR01-F-5.1-2	Shall update Decommuted_eng_mem_list with the values extracted from Current_HK_telem, Prior_HK_telem, and Sci_eng_data.	4.1-1
PR01-F-6.1	Shall close all Level 0 files upon completion of L1A granules.	5.1-3
PR01-F-6.2	Shall close all Level 0 files upon receipt of Program_stop.	5.1-3
PR01-F-6.3	Shall write Close_log_msgs to Log_messages when errors occur in the Close_processing_run process.	5.1-3
PR01-I-1	The L1A/Geolocation software shall follow the coding standards established by the MODIS project in MODIS Software Development Standards and Guidelines	3.3-1
PR01-I-2	The L1A/Geolocation software shall use all applicable mandatory SDPTK routines and any additional SDPTK routines that are useful.	2.3-1
PR01-I-3	The L1A/Geolocation software shall use HDF Version 4.0 for reading and writing all HDF files.	3.2-2

Req. #	Requirement	Traceability
PR01-S-1	The L1A and Geolocation software shall provide interfaces to mandatory SDPTK elements such as the SMF, PCF and the MCF.	2.3-5, 2.3-6
PR03-F-1-1	Shall read the Geolocation_Parameters.	ATBD 2.4
PR03-F-1-2	Shall report errors as parameter_input_status_messages in log status message file through the use of SDP functions.	5.2-1
PR03-F-1-3	Shall read the process_control_parameters from the SDP process control file as: Ephemeris_source.	CCR#309
PR03-F-2.1-1	Shall read the Level_1A_Data as level1a_global_metadata, level1a_scan_flags, spacecraft_ancillary_data, and instrument_scan_data.	ATBD 2.0, CCR#141
PR03-F-2.1-2	Shall generate data_input_status_messages if errors are encountered in reading Level_1A_Data.	5.2-1
PR03-F-2.2-1	Shall convert spacecraft_ancillary_data to converted_spacecraft_ancillary_data according to the GISS, using spacecraft_ancillary_data_conversion_parameters.	ATBD 3.1.4.2
PR03-F-2.3-1	Shall convert instrument_scan_data to converted_instrument_scan_data according to the specifications in CDRL 305, using the instrument_data_conversion_parameters.	ATBD 3.1.4.1
PR03-F-2.4-1	Shall detect corrupt converted_spacecraft_ancillary_data to produce validated_spacecraft_ancillary_data.	5.2-1, ATBD 3.1.4.2
PR03-F-2.4-2	Shall generate data_input_status_messages when corrupted data values are detected.	5.2-1
PR03-F-2.5-1	Shall detect corrupt converted_instrument_scan_data. converted_mirror_encoder_data.	5.2-1, ATBD 2.1
PR03-F-2.5-2	Shall generate data_input_status_messages and set scan_quality_flags if corrupted data values are detected.	5.2-1, 5.3-1
PR03-F-2.5-3	Shall produce validated_mirror_encoder_data. mirror_encoder_times from converted_instrument_scan_data. converted_mirror_encoder_data as described in the ATBD, section 3.1.4.1.	ATBD 3.1.4.1
PR03-F-2.5-4	Shall produce validated_mirror_encoder_data. mirror_encoder_positions from converted_instrument_scan_data. scan_mirror_start_positions as described in the ATBD, section 3.1.4.1.	ATBD 3.1.4.1
PR03-F-2.5-5	Shall incorporate level1a_scan_flags into scan_quality_flags.	5.3-1
PR03-F-2.6-1	Shall set the respective granule_science_state or granule_science_abnormal flags if either science_state or science_abnormal scan-level flags indicate that abnormal instrument or spacecraft state occurred.	CCR#309
PR03-F-3.1.1-1	Shall compute the time_of_observation for each frame from the instrument_timing, number_of_Earth_view_frames, and scan_start_times as defined in the Geolocation ATBD, section 3.1.4.1.	ATBD 3.1.4.1
PR03-F-3.1.1-2	Shall set values of spatial_element_quality_flags to indicate insufficient or invalid input data.	5.2-4.3, 5.3-1
PR03-F-3.1.2-1	Shall compute the mirror_scan_angle as defined in the Geolocation ATBD, section 3.1.4.1, from the time_of_observation, mirror_sides, mirror_model, and validated_mirror_encoder_data interpolated using a Chebyshev polynomial fit.	ATBD 3.1.4.1 (CCR#337)
PR03-F-3.1.2-2	Shall set spatial_element_quality_flags to indicate insufficient or invalid input data.	5.2-4.3, 5.3-1
PR03-F-3.1.3-1	Shall compute the mirror_normal_vector from the mirror_model, mirror_scan_angle, and mirror_sides as defined in the Geolocation ATBD, section 3.1.4.1.	ATBD 3.1.4.1
PR03-F-3.1.4-1	Shall compute lines_of_sight from mirror_normal_vector, focal_plane_geometry, and internal_coordinate_transformations as defined in the Geolocation ATBD, section 3.1.4.1.	ATBD 3.1.4.1

Req. #	Requirement	Traceability
PR03-F-3.2.1-1	Shall interpolate either <code>validated_spacecraft_ancillary_data</code> or <code>Spacecraft_Data</code> , as selected by <code>Ephemeris_source</code> , to the <code>time_of_observation</code> to determine <code>interpolated_spacecraft_data</code> as defined in the Geolocation ATBD, section 3.1.4.2.	ATBD 3.1.4.2
PR03-F-3.2.1-2	Shall generate <code>computation_status_messages</code> and set <code>spatial_element_quality_flags</code> in response to interpolation errors	5.2-1, 5.3-1
PR03-F-3.2.2-1	Shall transform the <code>interpolated_spacecraft_data</code> , and the <code>lines_of_sight</code> to the ECR reference frame as defined in the Geolocation ATBD, section 3.1.4.2.	ATBD 3.1.4.2
PR03-F-3.2.3-1	Shall determine <code>ellipsoid_intersection</code> using <code>lines_of_sight</code> and <code>spacecraft_orbit_position_ECR</code> as defined in the Geolocation ATBD, section 3.1.4.2.	ATBD 3.1.4.2
PR03-F-3.2.3-2	Shall generate <code>computation_status_messages</code> and set <code>spatial_element_quality_flags</code> in response to errors in determining <code>ellipsoid_intersection</code> .	5.2-1, 5.3-1
PR03-F-3.2.4-1	Shall compute the <code>geodetic_latitude</code> and <code>geodetic_longitude</code> from the <code>ellipsoid_intersection</code> as defined in the Geolocation ATBD, section 3.1.4.2.	ATBD 3.1.4.2
PR03-F-3.3.1-1	Shall compute the <code>terrain_intersection</code> of the <code>lines_of_sight_ECR</code> with the <code>terrain_height</code> for the <code>observed_locations</code> (<code>geodetic_latitude</code> and <code>geodetic_longitude</code>) as defined in the Geolocation ATBD, section 3.1.4.2.	ATBD 3.1.4.2
PR03-F-3.3.2-1	Shall compute the <code>observed_locations_with_terrain_correction</code> from the <code>terrain_intersection</code> as defined in the Geolocation ATBD, section 3.1.4.2.	ATBD 2.1
PR03-F-3.3.3-1	Shall read <code>Elevation_Data.digital_elevation_model_data</code> to determine <code>terrain_height</code> for region surrounding <code>observed_locations</code> .	ATBD 3.1.4.2
PR03-F-3.3.3-2	Shall set <code>spatial_element_quality_flags</code> to indicate <code>Elevation_Data</code> of inferior quality.	5.3-1
PR03-F-3.4.1-1	Shall determine the <code>vector_from_observed_locations_to_spacecraft_position</code> from the <code>observed_locations_with_terrain_correction</code> as defined in the Geolocation ATBD, section 3.1.4.2.	ATBD 3.1.4.2
PR03-F-3.4.2-1	Shall compute the <code>sensor_vectors</code> for the <code>observed_locations_with_terrain_corrections</code> from the <code>vector_from_observed_location_to_spacecraft_position</code> as defined in the Geolocation ATBD, section 3.1.34.2.	ATBD 2.1
PR03-F-3.4.2-2	Shall set <code>spatial_element_quality_flags</code> to identify unavailable or oversize sensor range.	CCR#343
PR03-F-3.4.3-1	Shall determine the <code>sun_reference_vector</code> in the ECR reference frame at the <code>time_of_observation</code> .	ATBD 3.1.4.2
PR03-F-3.4.4-1	Shall compute <code>solar_vectors</code> at the observed location from the <code>sun_reference_vector</code> and the <code>observed_locations_with_terrain_correction</code> as defined in the Geolocation ATBD, section 3.1.4.2.	ATBD 2.1
PR03-F-3.5.1-1	Shall interpolate <code>validated_spacecraft_ancillary_data</code> or <code>Spacecraft_Data</code> using the <code>solar_diffuser_start_times</code> and <code>number_of_solar_diffuser_frames</code> to determine <code>interpolated_spacecraft_data</code> .	CCR#309
PR03-F-3.5.2-1	Shall determine the <code>sun_reference_vector</code> in the ECR reference frame at the mid-point of the solar diffuser view sector using the <code>solar_diffuser_start_times</code> and the <code>number_of_solar_diffuser_frames</code> .	CCR#309
PR03-F-3.5.3-1	Shall determine the ECR reference frame-to-solar diffuser coordinate system transformation using the <code>interpolated_spacecraft_data</code> and the <code>internal_coordinate_transformations</code> .	CCR#309

Req. #	Requirement	Traceability
PR03-F-3.5.3-2	Shall transform the sun_reference_vector to the solar diffuser coordinate system.	CCR#309
PR03-F-3.5.3-3	Shall compute the solar_diffuser_angles from the transformed Sun vector.	CCR#309
PR03-F-3.6.1-1	Shall interpolate validated_space_ancillary_data or Spacecraft_Data using the space_view_start_times and number_of_space_view_frames to determine interpolated_spacecraft_data.	CCR#309
PR03-F-3.6.2-1	Shall determine the moon_reference_vector in the ECR reference frame at the mid-point of the space view sector using the space_view_start_times and the number_of_space_view_frames.	CCR#309
PR03-F-3.6.3-1	Shall determine the ECR reference frame-to-instrument coordinate system transformation using the interpolated_spacecraft_data and the internal_coordinate_transformations.	CCR#309
PR03-F-3.6.3-2	Shall transform the moon_reference_vector to the instrument coordinate system to determine the lunar_vector.	CCR#309
PR03-F-3.7-1	Shall determine Earth_View_Center_data as the spacecraft ephemeris and the sun reference vector at the Earth View's (odd) center frame's center time, as determined from number_of_Earth_view_frames, time_of_observation, interpolated_spacecraft_data, spacecraft_orbit_position_ECR, spacecraft_orbit_velocity_ECR and sun_reference_vector.	CCR#309
PR03-F-3.8-1	Shall determine the land-sea_mask_at_observed_locations at the observed_locations_with_terrain_correction from the land-sea_mask data.	CCR#319
PR03-F-4.1-1	Shall write the Geolocation Global Metadata deduced from the geolocation_scan_data, granule_level_state, and the level_1A_global_metadata as: geolocation_global_metadata.	4.2 (CCR#309)
PR03-F-4.1-2	Shall generate data_output_status_messages if write errors are encountered.	5.2-1
PR03-F-4.2-1	Shall write the focal_plane_geometry and geolocation_parameter_metadata to the Geolocation Product.	5.3-1 (CCR#122)
PR03-F-4.2-2	Shall generate data_output_status_messages if write errors are encountered.	5.2-1
PR03-F-4.3-1	Shall write the geolocation_scan_data as: product_fields_per_spatial_element and product_fields_per_scan.	ATBD 2.1
PR03-F-4.3-2	Shall generate data_output_status_messages if write errors are encountered.	5.2-1
PR03-F-4.3-3	Shall write the product_fields_per_spatial_element in HDF-EOS Swath format.	CCR#309
PR03-F-4.4-1	Shall write the Spatial Metadata deduced from the geolocation_scan_data and the level_1A_global_metadata to the L1A Product as: spatial_metadata.	CCR#148
PR03-F-4.4-2	Shall generate data_output_status_messages if write errors are encountered.	5.2-1
PR03-F-5.1-1	Shall read land_control_points and generate selected_land_control_points based on geographic boundaries represented by geolocation_scan_data and the control_point_exclusion_mask.	ATBD 2.1 (CCR#141)
PR03-F-5.1-2	Shall read ocean_control_points and generate selected_ocean_control_points based on geographic boundaries represented by geolocation_scan_data and the control_point_exclusion_mask.	ATBD 2.1 (CCR#141)

Req. #	Requirement	Traceability
PR03-F-5.2-1	Shall perform correlation of selected_land_control_points with land_control_point_radiances and geolocation_scan_data to generate land_control_point_matchups as specified in the Geolocation ATBD.	ATBD 2.1
PR03-F-5.2-2	Shall write land control point residuals data as: land_control_point_matchups.	CCR#310
PR03-F-5.3-1	Shall perform correlation of selected_ocean_control_points with ocean_control_point_radiances and geolocation_scan_data to generate ocean_control_point_matchups as specified in the Geolocation ATBD.	ATBD 2.1
PR03-F-5.3-2	Shall write ocean control point residuals data as: ocean_control_point_matchups.	CCR#337
PR03-F-5.4-1	Shall read the Level_1B_Data as level1B_global_metadata, level1B_scan_flags, and control_point_radiances.	ATBD 2.1 (CCR#141)
PR03-F-5.4-2	Shall generate data_input_status_messages if errors are encountered in reading Level_1B_Data.	5.2-1 (CCR#141)
PR03-F-5.5-1	Shall read the Level_2_Data for the MODIS Classification Masks (MOD35) and other MODIS Level 2 products that may contribute to the identification of regions in a granule that should be excluded for the purpose of geolocation process accuracy assessment as level2_global_metadata, level2_scan_flags, and L2_classification_masks.	ATBD 2.1 (CCR#141)
PR03-F-5.5-2	Shall generate data_input_status_messages if errors are encountered in reading Level_2_Data.	5.2-1 (CCR#141)
PR03-F-5.6-1	Shall determine the control_point_exclusion_mask of regions not to be used for control point correlation because of bad validation or quality of the data or because of cloud coverage or other scene distortion from the geolocation_scan_quality_flags and spatial_element_quality_flags, level1B_scan_flags, level1B_global_metadata, level_2_Classification_masks, level2_global_metadata, and level2_scan_flags.	4.4.3-3 ATBD 2.1 (CCR#141)
PR03-F-5.7-1	Shall write the control_point_metadata derived from control_point_parameters and geolocation_scan_data to each control point residuals file.	CCR#337
PR03-F-6.1-1	The L1A/Geolocation software shall generate initiation_messages at program commencement.	5.2-1 (CCR#278)
PR03-F-6.2-1	The L1A/Geolocation software shall generate termination_messages at program completion.	3.4-1, 3.5.2-2 (CCR#278)
PR03-P-1	The L1A/Geolocation products shall be produced within the data volume and processing load allocation shown in Table B-1 of the Science Data Processing Software Requirements Specification, Version 2 and Beyond, SDST-089.	3.2-2
PR03-I-1	The L1A/Geolocation software shall follow the coding standards established by the MODIS project in MODIS Software Development Standards and Guidelines	3.3-4
PR03-I-2	The L1A/Geolocation software shall use all applicable mandatory SDPTK 5.2 routines and any additional SDPTK routines that are useful.	3.3-4, 3.3-2
PR03-I-3	The L1A/Geolocation software shall use HDF Version 4.1r1 for reading and writing all HDF files.	4.1-2
PR03-I-4	The L1A/Geolocation software shall use HDF-EOS Version 2.0	3.3-4

Req. #	Requirement	Traceability
PR03-I-5	<p>The L1A/Geolocation software processes shall be able to process input data sets with the following nominal attributes:</p> <ul style="list-style-type: none"> • Time transitions, including Greenwich Meridian crossing, leap year, beginning of the year, beginning of the millennium, and/or start of data collection not on a granule boundary. • Terminator crossing, International Date Line crossing, and/or Earth pole within a granule. • Day, Night, or mixed mode data. 	5.2-3
PR03-I-6	<p>The L1A/Geolocation software processes shall handle input data sets with the following error attributes:</p> <ul style="list-style-type: none"> • Fill data, including filled pixels, filled bands, and/or filled scans. • Redundant data, consisting of repeated packets at Level 0 or duplicate input files at all product levels. • Corrupted data as indicated by Quality Assurance (QA) flags also located within the file. • Missing input MODIS product files, ancillary files, and look-up tables. • Wrong format input files. • Noisy or dead detectors, as indicated by QA flags located within the Level 1B file. 	5.2-4
PR03-S-1	The L1A and Geolocation software shall provide interfaces to mandatory SDPTK elements such as the SMF, PCF and the MCF.	3.3-3, 5.1-1
PR03-S-2	The L1A and Geolocation software shall utilize the NCSA product HDF to create all archive products.	3.2-2

(This page intentionally left blank.)

7. PARTITIONING FOR RELEASES

The delivery of MODIS L1A/Geolocation software has been partitioned into three releases; Beta, V1, and Version 2 (V2). The L1A/Geolocation software is being developed to meet the objectives for each delivery, as stated in Section 3.4.3 of the Team Leader Working Agreement (TLWA) for MODIS. The requirements contained within this document apply to the V2 software delivery.

(This page intentionally left blank.)

8. GLOSSARY/SYMBOLS

The following symbols are used as part of the requirements model. The use of these symbols is consistent with the use specified by Yourdon and Demarco, as implemented by the Cadre Technologies, Inc. CASE tool *Teamwork*.

Process - A process is represented by a circle on a DFD. A process may consist of other processes (a 'parent' process) or it may be the lowest level of decomposition (a 'child' process).



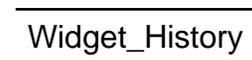
Terminator - A terminator, shown as a box on a context diagram, represents an entity which exists outside of the system that communicates with the system, but cannot be otherwise altered by the system.



Data Flow - A data flow is shown as a solid line on a DFD. Data moves in the direction of the arrow, and the content of the data is given by the flow's name.



Data Store - A data store is shown as two horizontal solid lines on a DFD. A data store represents information that must be remembered. The store is named to reflect what data is kept within it.



Control Flow - Control flows are represented on a DFD by a dashed line. A control flow is an event, condition, or signal that causes specific actions to be taken. Control flows must terminate at a c-spec.



Control Specification (c-spec) - C-specs are shown as a vertical line on a DFD. C-specs represent the actions that must be taken in response to a control flow.



The following symbols are used as part of the data dictionary:

<u>Operator</u>	<u>Example</u>	<u>Meaning</u>
+	a + b	a together with b
[]	[a b]	either a or b
{ }	1{a}10	1 to 10 instances of a (defaults: lower bound 0, upper bound infinity)
**	*comments*	comment delimiter

The following terminology is used to within the data dictionary to describe the lowest-level elements:

- **cel** - short for continuous element, this is used when an element is a real number.
- **del** - short for discrete element, this is used when an element is an integer or an enumeration.
- **pel** - short for primitive element, this is used when an element is a string or a structure.

APPENDIX A: ACRONYMS

AEM	Analog Electronics Modules
APID	Application Identifier
ATBD	Algorithm Theoretical Basis Document
CASE	Computer Aided Software Engineering
CCSDS	Consultive Committee for Space Data Systems
CDRL	Contract Deliverable Requirements List
CP	Control Point or Command and Telemetry Processor
DCR	Direct Current Restore
DDE	Data Dictionary Entries
DEM	Digital Elevation Model
DFD	Data Flow Diagram
ECR	Earth-Centered Rotating
ECI	Earth-Centered Inertial
ECS	EOS Core System
EDOS	EOSDIS Data and Operations System
EOS	Earth Observing System
EOSDIS	Earth Observing System Data and Information System
EV	Earth View
FAM	Forward-viewing Analog electronics Module
FPA	Focal Plane Arrays
FR	Formatter (component of MODIS instrument)
GIIS	General Instrument Interface Specification
GSFC	Goddard Space Flight Center
HDF	Hierarchical Data Format
L1A	Level 1A
LOS	Line of Sight
MCF	Metadata Configuration File
MODIS	Moderate Resolution Imaging Spectroradiometer
NASA	National Aeronautics and Space Administration
NCSA	National Center for Supercomputing Applications
p-specs	Process Specifications
PCF	Process Control File
PGE	Product Generation Executive
QA	Quality Assurance
QC	Quality Control
SA/RT	Structured Analysis with Real Time Extensions
SBRS	Santa Barbara Remote Sensing Center
SDP	Science Data Production

SDPTK	Science Data Production Toolkit
SDST	Science Data Support Team
SIG	Silicon Graphics Inc.
SMF	Status Message Facility
TAI	International Atomic Time
TBD	To Be Determined
TLCF	Team Leader Computing Facility
TLWA	Team Leader Working Agreement
TM	Technical Memorandum or Thematic Mapper
V1	Version 1
V2	Version 2

APPENDIX B: LEVEL 1A DATA FLOW DIAGRAMS

This EPS image does not contain a screen preview.

It will print correctly to a PostScript printer.

File Name : Figure_B1.EPSF

Title : Untitled

Creator : Cadre Printing

CreationDate : Thu 11 Sep 1997 19:10:32 EDT

Pages : 0

Figure B-1. MODIS Level 1A Processing

Table B-1. Control Activation

	“Write_ failed_ packet”	“Read_ MODIS_ packet”	“Unpack_ packet_ contents”	“Close_ processing_ run”
Program_stop				1
Abort_current_packet_processing	1	2		

**This EPS image does not contain a screen preview.
It will print correctly to a PostScript printer.
File Name : Figure_B2.EPSF
Title : Untitled
Creator : Cadre Printing
CreationDate : Thu 11 Sep 1997 19:12:24 EDT
Pages : 0**

Figure B-2. Initialize_Level_1A_processing

**This EPS image does not contain a screen preview.
It will print correctly to a PostScript printer.**

File Name : Figure_B3.EPSF

Title : Untitled

Creator : Cadre Printing

CreationDate : Thu 11 Sep 1997 19:12:59 EDT

Pages : 0

Figure B-3. Process_Packets

This EPS image does not contain a screen preview.

It will print correctly to a PostScript printer.

File Name : Figure_B4.EPSF

Title : Untitled

Creator : Cadre Printing

CreationDate : Thu 11 Sep 1997 19:13:47 EDT

Pages : 0

Figure B-4. Verify_MODIS_packet

**This EPS image does not contain a screen preview.
It will print correctly to a PostScript printer.**

File Name : Figure_B6.EPSF

Title : Untitled

Creator : Cadre Printing

CreationDate : Thu 11 Sep 1997 19:14:50 EDT

Pages : 0

Figure B-5. Create_MODIS_Scans

**This EPS image does not
It will print correctly to
File Name : Figure_B7.EF
Title : Untitled
Creator : Cadre Printing
CreationDate : Thu 11 S
Pages : 0**

Figure B-6. Create_decommutated_eng_mem_list

APPENDIX C: LEVEL 1A DATA DICTIONARY

This appendix contains the data dictionary elements that support the L1A requirements model. The appendix is divided into two sections: alphabetical and by external entities.

C.1 Alphabetic Listing

This section is an alphabetic listing of all elements within the data dictionary. If a data dictionary entry is a composite of other data dictionary entries, then the entry shows the relationships to the other data dictionary entries using the notation found in the Glossary/Symbols section. If a data dictionary entry is not composed of other data dictionary entries, then that data dictionary entry shows the entry's type, range, and description, and the entry is labeled as either cel, del or pel (see Glossary/Symbols sections for explanation).

Abort_current_packet_processing (control flow, pel) = *

Type: Event

Range: N/A

Description: Self-explanatory. *

Cal_frame_count (data flow, del) = *

Type: Integer

Range: 0 - 64

Description: This is the frame count for calibration scenes, found in packet bits 126-131. *

CCSDS_application_ID (data flow, del) = *

Type: Integer

Range: 64 - 127 (only 64 is expected to be used)

Description: This is the CCSDS APID field, found in packet bits 5-15. The EOS-AM spacecraft has assigned the MODIS instrument APIDs 64 through 127; MODIS plans to use only one APID per instrument at this point (for EOS-AM, this will be 64). *

CCSDS_mirror_side(data flow, del) = *

Type: Enumeration

Range: 0 or 1 (Side 1 or Side 2)

Description: This is the CCSDS mirror side indicator (packet bit 119). *

CCSDS_packet_count (data flow, del) = *

Type: Integer

Range: 0 - 16383

Description: This is the packet count found in the CCSDS header, packet bits 18-31. *

CCSDS_packet_length (data flow, del) = *

Type: Integer

Range: 0 - 65535

Description: This is the CCSDS packet length field, found in packet bits 32-47. For MODIS, this field should contain 635 for long packets, and 269 for short packets. *

CCSDS_packet_sequence_flag (data flow, del) = *

Type: Enumeration

Range: 0, 1, 2, or 3 (Middle, first, last, or unsegmented)

Description: This is the packet sequence indicator from the CCSDS header, bits 16-17. MODIS short packets are unsegmented; MODIS long packets have a first and last segment. There are no middle segments in MODIS data. *

CCSDS_packet_type (data flow, del) = *

Type: Enumeration

Range: 0, 1, 2, 3, 4, 5 through 7 (Day, Night, Eng1, Test, Eng2, Spare)

Description: This is the CCSDS packet type indicator (packet bits 113-115). *

CCSDS_primary_header (data flow) =

CCSDS_version

+ CCSDS_type

+ CCSDS_secondary_header_flag

+ CCSDS_application_ID

+ CCSDS_packet_sequence_flag

+ CCSDS_packet_count

+ CCSDS_packet_length

CCSDS_quick_look(data flow, del) = *

Type: Enumeration

Range: 0 or 1 (QL sel or QL not sel)

Description: This is the CCSDS quick look indicator (packet bit 112). *

CCSDS_scan_count (data flow, del) = *

Type: Integer

Range: 0 through 7

Description: This is the CCSDS scan count (packet bits 116-118) that relates data to a particular near term scan. *

CCSDS_secondary_header (data flow) = *

CCSDS_time_tag

+ CCSDS_quick_look

+ CCSDS_packet_type

+ CCSDS_scan_count

+ CCSDS_mirror_side

CCSDS_secondary_header_flag (data flow, del) = *

Type: Enumeration

Range: 0 or 1

Description: This is the CCSDS secondary header flag; a value of 1 indicates that a secondary header is present, a value of 0 indicates that there is no secondary header present. For EOS-AM, this should always be a 1. *

CCSDS_time_tag (data flow, del) = *

Type: Structure

Range: N/A

Description: This is the MODIS packet time code (packet bits 48-111); for EOS-AM, this is a variant of the CCSDS Day Segment format. *

CCSDS_type (data flow, del) = *

Type: Enumeration

Range: 0 or 1

Description: This is a 1-bit field (bit 3) in the CCSDS header. For MODIS, this field should always be 0. *

CCSDS_version (data flow, del) = *

Type: Enumeration

Range: 0 - 7

Description: This is the CCSDS packet version flag (packet bits 0-2). For MODIS, this field must be 0. *

Close_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Close_processing_run processes. *

Command Parameters (data flow, pel) = *

Type: Structure

Range: N/A

Description: This is the command parameter data; a block of 320 bits containing fields of varying bit size. *

Current_HK_telem (data flow, pel) = *

Type: Structure

Range: N/A

Description: This is the latest MODIS housekeeping telemetry block that was passed to the spacecraft. The structure is always 512 bits in size, but the exact contents vary depending upon the spacecraft's major cycle. *

Current_SC_ancill_data (data flow, pel) = *

Type: Structure

Range: N/A

Description: This is the current spacecraft ancillary data; this 512 bit structure is unpacked by geolocation. *

Decommutated_eng_mem_list (data flow, pel) = *

Type: Structure

Range: N/A

Description: This is the list of all engineering and memory data output by the instrument; the values represent the last values received for the respective engineering or memory data. *

Earth_encoder_time (data flow, del) = *

Type: Integer

Range: 0 - 65535

Description: This is the time between every 100th mirror encoder pulse over the earth scene, expressed in microseconds. Nominally, this value should be 18032. *

Earth_frame_count (data flow, del) = *

Type: Integer

Range: (theoretical) 0 - 2048 (practical) 0 - 1400

Description: This is the Earth frame count, extracted from packet bits 121-131 when the frame type flag (bit 120) is equal to 0. *

ECS_standard_global_metadata (data flow, pel) = *

Type: Structure

Range: N/A

Description: This is the structure containing the ECS Standard global metadata used in the L1A product. *

Eng1_Pkt1_fill (data flow, pel) = *

Type: N/A

Range: N/A

Description: This is a field of 580 bits of fill at the end of the MODIS engineering group 1 packet #1 data field. *

Eng1_Pkt2_fill (data flow, pel) = *

Type: N/A

Range: N/A

Description: This is a field of 1012 bits of fill at the end of the MODIS engineering group 1 packet #2 data field. *

Eng2_Pkt1_fill (data flow, pel) = *

Type: N/A

Range: N/A

Description: This is a field of 2612 fill bits at the end of the MODIS engineering group 2 packet #1 data field. *

Eng2_Pkt2_fill (data flow, pel) = *

Type: N/A

Range: N/A

Description: This is a field of 580 fill bits at the end of the MODIS engineering group 2 packet #2 data field. *

Eng1_pkt_1 (data flow) =

550 { FPA_DCR_offset

} 550

+ Eng1_Pkt1_fill

Eng1_pkt_2 (data flow) =
78 { Earth_encoder_time
} 78
+ 40 { View_sector_definitions
} 40
+ 24 { View_sector_actuals
} 24
+Sci_eng_data
+ Eng1_Pkt2_fill

Eng2_pkt_1 (data flow) =
Current_HK_telem
+ Prior_HK_telem
+ Current_SC_ancill_data
+ Prior_SC_ancill_data
+ 40 { Command_Parameters
} 40
+ Eng2_Pkt1_fill

Eng2_pkt_2 (data flow) =
550 { PV_Gains
} 550
+ Eng2_Pkt2_fill

Eng_Mem_data (data flow) =
[Unpacked_eng1_packet_1
1 Unpacked_eng1_packet_2
1 Unpacked_eng2_packet_1
1 Unpacked_eng2_packet_2]

Failed_pkts (store) =
(MODIS_pkt)

FPA-AEM_config (data flow, del) = *

Type: Array

Range: 0 or 1

Description: This is a array of 10 1-bit flags that indicate which Focal Plane Arrays (FPA) are controlled by which Analog Electronics Modules (AEM). This field is found at packet bits 132-141; it is assumed that all packets within the same scan will contain the same values for these 10 bits. *

FPA_DCR_offset (data flow, del) = *

Type: Integer

Range: 0 - 255

Description: This is the DC Restore (DCR) offset; there is one per scan for each detector. The values are found in Engineering packet 1. *

Frame_nightmode_data (data flow) =
 10 { 17 { Unpacked_MODIS_radiance
 } 17
 } 10

Half_frame_daymode_data (data flow) =
 5 { 2 { 16 { Unpacked_MODIS_radiance
 } 16
 } 2
 + 5 { 4 { Unpacked_MODIS_radiance
 } 4
 } 5
 + 14 { Unpacked_MODIS_radiance
 } 14
 + 17 { Unpacked_MODIS_radiance
 } 17
 } 5

Init_log_msgs (data flow) =
 Level0_open_log_msgs
 + Level0_header_log_msgs

L1A_product_period (data flow, del) = *
 Type: Constant
 Range: 5.0 minutes +/- error term (small)
 Description: This is the temporal duration of an average single L1A_product. *

Level0_data (data flow) =
 Level0_file_header +
 MODIS_packets

Level0_end_time (data flow, cel) = *
 Type: Double
 Range: Jan. 1, 1958 - End of mission
 Description: This is the field containing the end time for the Level 0 file. *

Level0_file_header (data flow) =
 Num_packets_in_file +
 Level0_start_time +
 Level0_end_time

Level0_header_log_msgs (data flow, pel) = *
 Type: text
 Range: N/A
 Description: These are the status messages generated by the
 Verify_Level0_file_header process. *

Level0_open_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Open_Level0_file process. *

Level0_open_metadata (data flow, pel) = *

Type: Structure

Range: N/A

Description: This structure contains metadata obtained from opening the Level0_data file. *

Level0_start_time (data flow, cel) = *

Type: Double

Range: Jan 1, 1958 - End of mission

Description: This is the field containing the start time for the Level 0 file. *

Level1A_data (data flow) =

ECS_standard_global_metadata

+ MODISL1A_specific_global_metadata

+ 1 { MODIS_scan }

+ Failed_pkts

Level1A_Parameters (data flow) =

Spacecraft_indicator

+ Processing_mode

Log_messages (data flow) =

Init_log_msgs

+ Packet_log_msgs

+ Scan_log_msgs

+ Product_log_msgs

+ Close_log_msgs

Missing_Packet_log_msgs (data flow, pel) = *

Type: Text

Range: N/A

Description: these are status messages generated by the Process_MODIS_packet process. *

MODIS_header (data flow) =

Packet_frame_type

+ [Earth_frame_count

| Packet_cal_type

+ Packet_cal_mode

+ Cal_frame_count]

+ FPA-AEM_config

+ Science_state

+ Science_abnormal

MODIS_packets (data flow) =
1{MODIS_pkt}

MODIS_pkt (data flow) =
CCSDS_primary_header
+ CCSDS_secondary_header
+ MODIS_header
+ Packet_data_field
+ Packet_checksum

MODIS_scan (data flow) =
MODIS_scan_radiances +
MODIS_scan_level_metadata +
MODIS_scan_pixel_quality +
Decommutated_eng_mem_list

MODIS_scan_data (store) =
MODIS_scan_radiances
+ MODIS_scan_level_metadata
+ Scan_eng_mem_data

MODIS_scan_level_metadata (data flow, pel) = *
Type: Structure
Range: N/A
Description: This is the structure which contains the various MODIS scan-level metadata variables referenced in the MODIS L1A product specification. *

MODIS_scan_period (data flow, del) = *
Type: Constant
Range: 1.4771 +/- error term (small)
Description: This is the period of one half-rotation of the MODIS scan mirror. *

MODIS_scan_pixel_quality (store, pel) = *
Type: Structure
Range: N/A
Description: This is the structure containing the pixel quality data for each scan. *

MODIS_scan_radiances (data flow) =
[Half_frame_daymode_data
| Frame_nightmode_data]

MODISL1A_specific_global_metadata (data flow, pel) = *
Type: Structure
Range: N/A
Description: This is the structure containing the global metadata in the L1A product that is not part of the ECS Standard metadata definition. *

Num_packets_in_file (data flow, del) = *

Type: Integer

Range: 1 - ?

Description: This is the field containing the number of packets found in the Level 0 file. *

Packet_cal_mode (data flow, del) = *

Type: Enumeration

Range: 0, 1, or 2 (Radiometric, Spatial, or Spectral calibration mode)

Description: This is the calibration mode (for the SRCA?), found in packet bits 123-124 when the frame type flag (bit 120) is 1. *

Packet_cal_type (data flow, del) = *

Type: Enumeration

Range: 0, 1, 2, or 3 (SD, SRCA, BB, or SV)

Description: This is the calibration type flag, found in packet bits 121-122 when the frame type flag (bit 120) is 1. *

Packet_checksum (data flow, cel) = *

Type: Continuous

Range: x000 - xfff

Description: This is the 12 bit checksum field for the packet, found in the last 12 bits of the packet (bits 5124-5135 for the long packets, bits 2196-2207 for the short packets). *

Packet_checksum_QC_data (data flow, pel) = *

Type: Structure

Range: N/A

Description: This is the structure that contains the QC information pertaining to the checksum checking process. *

Packet_data_field (data flow) =

```
[ Packet_daymode_radiance_data
| Packet_nightmode_radiance_data
| Eng1_pkt_1
| Eng1_pkt_2
| Eng2_pkt_1
| Eng2_pkt_2 ]
```

Packet_daymode_radiance_data (data flow) =

```
415 { Packet_raw_radiance
} 415
```

Packet_frame_count (data flow) =

```
[Earth_frame_count | Cal_frame_count]
```

Packet_frame_type (data flow, del) = *

Type: Enumeration

Range: 0 or 1 (Earth or Calibration)

Description: This field (found at packet bit 120) is basically a flag which indicates how to interpret the next 11 bits. *

Packet_header_QC_data (data flow) =
CCSDS_header_QC_data
+ MODIS_header_QC_data
+ Packet_checksum_QC_data

Packet_log_msgs (data flow) =
Read_packet_log_msgs
+ Verify_packet_log_msgs
+ Packet_sequence_log_msgs
+ Dropout_log_msgs

Packet_mirror_side (data flow, del) = *
Type: Enumeration
Range: 0 or 1
Description: This field (found at packet bit 119) indicates which side of the MODIS mirror was used for the current scan. *

Packet_MODIS_type (data flow, del) = *
Type: Enumeration
Range: 0, 1, 2, 3, or 4 (Day, Night, Engineering, Test, or Memory Dump)
Description: This field indicates what type of MODIS data is stored in the current packet. This field is found at packet bits 113-115. *

Packet_nightmode_radiance_data (data flow) =
171 { Packet_raw_radiance
} 171

Packet_raw_radiance (data flow, del) = *
Type: 12-bit integer
Range: 0 - 4095
Description: This is the instantaneous output of the MODIS detectors as found in the MODIS packet. *

Prior_HK_telem (data flow, pel) = *
Type: Structure
Range: N/A
Description: This is the prior MODIS housekeeping telemetry block that was passed to the spacecraft. The structure is always 512 bits in size, but the exact contents vary depending upon the spacecraft's major cycle. *

Prior_SC_ancill_data (data flow, pel) = *
Type: Structure
Range: N/A
Description: This is the prior spacecraft ancillary data; this 512 bit structure is unpacked by geolocation. *

Processing_mode (data flow, del) = *

Type: Enumeration

Range: Normal or Expedited

Description: This is the parameter which indicates under which mode the current processing is run. *

Product_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Create_Level_1A_Products processes. *

Program_stop (control flow, pel) = *

Type: Event

Range: N/A

Description: This is a stop signal which will initiate a controlled shutdown of the program. *

PV_Gains (data flow, del) = *

Type: Structure

Range: N/A

Description: This is the structure containing the PV Gains from Engineering group 2 packet #2. *

Read_packet_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Read_MODIS_packets process. *

Scan_eng_mem_data (data flow) =

Decommutated_eng_mem_list

+ Eng_Mem_data

Scan_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Create_MODIS_Scans processes. *

Science_abnormal (data flow) = *

Type: Enumeration

Range: 0 or 1 (abnormal or normal)

Description: This field is the science abnormal indicator (packet bit 143)

Science_state (data flow) = *

Type: Enumeration

Range: 0 or 1 (test or normal)

Description: This field is the science state indicator (packet bit 142)

Sci_eng_data (data flow, pel) = *

Type: Structure

Range: N/A

Description: This is the MODIS science and engineering data from engineering packet 1, a 1536 bit structure containing fields of varying bit size. *

Spacecraft_indicator (data flow, del) = *

Type: Enumeration

Range: EOS-AM or EOS-PM

Description: This is the parameter which indicates which spacecraft's data is expected to be processed during this processing run. *

Unpacked_eng1_packet_1 (data flow) =

```
550 { FPA_DCR_offset
} 550
```

Unpacked_eng1_packet_2 (data flow) =

```
78 { Earth_encoder_time
} 78
+ 40 { View_sector_definitions
} 40
+ 24 { View_sector_actuals
} 24
+ Sci_eng_data
```

Unpacked_eng2_packet_1 (data flow) =

```
Current_HK_telem
+ Prior_HK_telem
+ Current_SC_ancill_data
+ Prior_SC_ancill_data
+ 40 { Command_Parameters
} 40
```

Unpacked_eng2_packet_2 (data flow) =

```
550 { PV_Gains
} 550
```

Unpacked_MODIS_radiance (data flow, del) = *

Type: Integer (16-bit)

Range: -32768 - 32767

Description: This is the instantaneous output of a MODIS detector, unpacked into 16 bits. *

Unpacked_radiance_packet_data (data flow) =

```
[ 415 { Unpacked_MODIS_radiance
} 415
| 171 { Unpacked_MODIS_radiance
} 171 ]
```

Verify_CCSDS_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Verify_CCSDS_header_info process. *

Verify_checksum_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Verify_packet_checksum process. *

Verify_MODIS_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Verify_MODIS_header_info process. *

Verify_packet_log_msgs (data flow) =

Verify_CCSDS_log_msgs

+ Verify_MODIS_log_msgs

+ Verify_checksum_log_msgs

View_sector_actuals (data flow) = *

Type: N/A

Range: N/A

Description: This is the view sector actuals from the Engineering group 1 packet #2. *

View_sector_definitions (data flow) = *

Type: N/A

Range: N/A

Description: This is the view sector definitions from the Engineering group 1 packet #2. *

C.2 Major Structures Decomposed

This second section of the data dictionary is organized by external entities shown on the model's context diagram (Figure 4-1). Each entity is listed with its complete decomposition. It is hoped that this alternate arrangement of data dictionary entries will provide a straightforward way for readers to understand the specific information contained in the external interfaces.

C.2.1 Level0_data

Level0_data (data flow) =

Level0_file_header +
MODIS_packets

Level0_file_header (data flow) =

Num_packets_in_file +
Level0_start_time +
Level0_end_time

Num_packets_in_file (data flow, del) = *

Type: Integer

Range: 1 - ?

Description: This is the field containing the number of packets found in the Level 0 file. *

Level0_start_time (data flow, cel) = *

Type: Double

Range: Jan 1, 1958 - End of mission

Description: This is the field containing the start time for the Level 0 file. *

Level0_end_time (data flow, cel) = *

Type: Double

Range: Jan. 1, 1958 - End of mission

Description: This is the field containing the end time for the Level 0 file. *

MODIS_packets (data flow) =

1 {MODIS_pkt}

MODIS_pkt (data flow) =

CCSDS_primary_header
+ CCSDS_secondary_header
+ MODIS_header
+ Packet_data_field
+ Packet_checksum

CCSDS_primary_header (data flow) =

CCSDS_version
+ CCSDS_type
+ CCSDS_secondary_header_flag
+ CCSDS_application_ID
+ CCSDS_packet_sequence_flag
+ CCSDS_packet_count
+ CCSDS_packet_length

CCSDS_version (data flow, del) = *

Type: Enumeration
Range: 0 - 7
Description: This is the CCSDS packet version flag (packet bits 0-2). For MODIS, this field must be 0. *

CCSDS_type (data flow, del) = *

Type: Enumeration
Range: 0 or 1
Description: This is a 1-bit field (bit 3) in the CCSDS header. For MODIS, this field should always be 0. *

CCSDS_secondary_header_flag (data flow, del) = *

Type: Enumeration
Range: 0 or 1
Description: This is the CCSDS secondary header flag; a value of 1 indicates that a secondary header is present, a value of 0 indicates that there is no secondary header present. For EOS-AM, this should always be a 1. *

CCSDS_application_ID (data flow, del) = *

Type: Integer
Range: 64 - 127
(only 64 is expected to be used)
Description: This is the CCSDS APID field, found in packet bits 5-15. The EOS-AM spacecraft has assigned the MODIS instrument APIDs 64 through 127; MODIS plans to use only one APID per instrument at this point (for EOS-AM, this will be 64). *

CCSDS_packet_sequence_flag (data flow, del) = *

Type: Enumeration
Range: 0, 1, 2, or 3
(Middle, first, last, or unsegmented)
Description: This is the packet segmentation indicator from the CCSDS header, bits 16-17. MODIS short packets are unsegmented; MODIS long packets have a first and last segment. There are no middle segments in MODIS data. *

CCSDS_packet_count (data flow, del) = *

Type: Integer

Range: 0 - 16383

Description: This is the packet count found in the CCSDS header, packet bits 18-31. *

CCSDS_packet_length (data flow, del) = *

Type: Integer

Range: 0 - 65535

Description: This is the CCSDS packet length field, found in packet bits 32-47. For MODIS, this field should contain 635 for long packets, and 269 for short packets. *

CCSDS_secondary_header (data flow, cel) = *

CCSDS_time_tag

+ CCSDS_quick_look

+ CCSDS_packet_type

+ CCSDS_scan_count

+ CCSDS_mirror_side

CCSDS_time_tag (data flow, del) = *

Type: Structure

Range: N/A

Description: This is the MODIS packet time code (packet bits 48-111); for EOS-AM, this is a variant of the CCSDS Day Segment format. *

CCSDS_quick_look(data flow, del) = *

Type: Enumeration

Range: 0 or 1 (QL sel or QL not sel)

Description: This is the CCSDS quick look indicator (packet bit 112). *

CCSDS_packet_type (data flow, del) = *

Type: Enumeration

Range: 0, 1, 2, 3, 4, 5, 6, 7 (Day, Night, Eng1, Test, Eng2, Spare)

Description: This is the CCSDS packet type indicator (packet bits 113-115). *

CCSDS_scan_count (data flow, del) = *

Type: Integer

Range: 0 through 7

Description: This is the CCSDS scan count (packet bits 116-118) that relates data to a particular near term scan. *

CCSDS_mirror_side(data flow, del) = *

Type: Enumeration

Range: 0 or 1 (Side 1 or Side 2)

Description: This is the CCSDS mirror side indicator (packet bit 119). *

MODIS_header (data flow) =

Packet_frame_type
 + [Earth_frame_count
 | Packet_cal_type
 + Packet_cal_mode
 + Cal_frame_count]
 + FPA-AEM_config
 + Science_state
 + Science_abnormal

Packet_frame_type (data flow, del) = *

Type: Enumeration

Range: 0 or 1 (Earth or Calibration)

Description: This field (found at packet bit 120) is basically a flag which indicates how to interpret the next 11 bits. *

Earth_frame_count (data flow, del) = *

Type: Integer

Range: (theoretical) 0 - 2048
 (practical) 0 - 1400

Description: This is the Earth frame count, extracted from packet bits 121-131 when the frame type flag (bit 120) is equal to 0. *

Packet_cal_type (data flow, del) = *

Type: Enumeration

Range: 0, 1, 2, or 3 (SD, SRCA, BB, or SV)

Description: This is the calibration type flag, found in packet bits 121-122 when the frame type flag (bit 120) is 1. *

Packet_cal_mode (data flow, del) = *

Type: Enumeration

Range: 0, 1, or 2 (Radiometric, Spatial, or Spectral calibration mode)

Description: This is the calibration mode (for the SRCA), found in packet bits 123-124 when the frame type flag (bit 120) is 1. *

Cal_frame_count (data flow, cel) = *

Type: Integer

Range: 0 - 64

Description: This is the frame count for calibration scenes, found in packet bits 126-131. *

FPA-AEM_config (data flow, del) = *

Type: Array

Range: 0 or 1

Description: This is a array of 10 1-bit flags that indicate which FPA are controlled by which AEM. This field is found at packet bits 132-141; it is assumed that all packets within the same scan will contain the same values for these 10 bits. *

Science_state (data flow) = *

Type: Enumeration

Range: 0 or 1 (test or normal)

Description: This field is the sciene state indicator (packet bit 142)

Science_abnormal (data flow) = *

Type: Enumeration

Range: 0 or 1 (abnormal or normal)

Description: This field is the sciene abnormal indicator (packet bit 143)

Packet_data_field (data flow) =

```
[ Packet_daymode_radiance_data
| Packet_nightmode_radiance_data
| Eng1_pkt_1
| Eng1_pkt_2
| Eng2_pkt_1
| Eng2_pkt_2 ]
```

Packet_daymode_radiance_data (data flow) =

```
415 { Packet_raw_radiance
} 415
```

Packet_raw_radiance (data flow, del) = *

Type: 12-bit integer

Range: 0 - 4095

Description: This is the instantaneous output of the MODIS detectors as found in the MODIS packet. *

Packet_nightmode_radiance_data (data flow) =

```
171 { Packet_raw_radiance
} 171
```

Packet_raw_radiance (data flow, del) = *

Type: 12-bit integer

Range: 0 - 4095

Description: This is the instantaneous output of the MODIS detectors as found in the MODIS packet. *

Eng1_pkt_1 (data flow) =

```
550 { FPA_DCR_offset
} 550
+ Eng1_pkt1_fill
```

FPA_DCR_offset (data flow, del) = *

Type: Integer

Range: 0 - 255

Description: This is the DCR offset; there is one per scan for each detector. The values are found in Engineering packet 1. *

Eng1_pkt1__fill (data flow, del) = *

Type: N/A

Range: N/A

Description: This is a field of 580 bits of fill at the end of the MODIS engineering group 1 packet #1 data field. *

Eng1_pkt_2 (data flow) =

78 { Earth_encoder_time

} 78

+ 40 { View_sector_definitions

} 40

+ 24 { View_sector_actuais

} 24

+ Sci_eng_data

+ Eng1_pkt2_fill

Earth_encoder_time (data flow, del) = *

Type: Integer

Range: 0 - 65535

Description: This is the time between every 100th mirror encoder pulse over the earth scene, expressed in microseconds. Nominally, this value should be 18032. *

View_sector_definitions (data flow) = *

Type: N/A

Range: N/A

Description: This is the view sector definitions from the Engineering group 1 packet #2. *

View_sector_actuais (data flow) = *

Type: N/A

Range: N/A

Description: This is the view sector actuals from the Engineering group 1 packet #2. *

Sci_eng_data (data flow, pel) = *

Type: Structure

Range: N/A

Description: This is the MODIS science and engineering data from engineering packet 1, a 1536 bit structure containing fields of varying bit size. *

Eng1_pkt2_fill (data flow, pel) = *

Type: N/A

Range: N/A

Description: This is a field of 1012 fill bits at the end of the MODIS engineering group 1 packet #2 data field. *

Eng2_pkt_1 (data flow) =
 Current_HK_telem
 Prior_HK_telem
 + Current_SC_ancill_data
 + Prior_SC_ancill_data
 + 40 { Command_Parameters
 } 40
 + Eng2_Pkt1_fill

Current_HK_telem (data flow, pel) = *
 Type: Structure
 Range: N/A
 Description: This is the latest MODIS housekeeping telemetry block that was passed to the spacecraft. The structure is always 512 bits in size, but the exact contents vary depending upon the spacecraft's major cycle. *

Prior_HK_telem (data flow, pel) = *
 Type: Structure
 Range: N/A
 Description: This is the prior MODIS housekeeping telemetry block that was passed to the spacecraft. The structure is always 512 bits in size, but the exact contents vary depending upon the spacecraft's major cycle. *

Current_SC_ancill_data (data flow, pel) = *
 Type: Structure
 Range: N/A
 Description: This is the current spacecraft ancillary data; this 512 bit structure is unpacked by geolocation. *

Prior_SC_ancill_data (data flow, pel) = *
 Type: Structure
 Range: N/A
 Description: This is the prior spacecraft ancillary data; this 512 bit structure is unpacked by geolocation. *

Command_parameters (data flow, pel) = *
 Type: Structure
 Range: N/A
 Description: This is the command parameter data, a block of 320 bits containing fields of varying bit size. *

Eng2_pkt1_fill (data flow, pel) = *
 Type: N/A
 Range: N/A
 Description: This is a field of 2612 bits of fill at the end of the MODIS Engineering group 2 packet #1 data field. *

Eng2_pkt_2 (data flow) =
 550 { PV_Gains
 } 550
 + Eng2_pkt2_fill

PV_Gains (data flow, del) = *
 Type: Structure
 Range: N/A
 Description: This is the structure containing the PV Gains from Engineering group 2 packet #2. *

Eng2_pkt2_fill (data flow, pel) = *
 Type: N/A
 Range: N/A
 Description: This is a field of 580 bits of fill at the end of the MODIS Engineering group 2 packet #2 data field. *

Packet_checksum (data flow, cel) = *
 Type: Continuous
 Range: x000 - xfff
 Description: This is the 12 bit checksum field for the packet, found in the last 12 bits of the packet (bits 5124-5135 for the long packets, bits 2196-2207 for the short packets). *

C.2.2 Level1A_data

Level1A_data (data flow) =
 ECS_standard_global_metadata
 + MODISL1A_specific_global_metadata
 + 1 { MODIS_scan }

ECS_standard_global_metadata (data flow, pel) = *
 Type: Structure
 Range: N/A
 Description: This is the structure containing the ECS Standard global metadata used in the L1A product. *

MODISL1A_specific_global_metadata (data flow, pel) = *
 Type: Structure
 Range: N/A
 Description: This is the structure containing the global metadata in the L1A product that is not part of the ECS Standard metadata definition. *

MODIS_scan (data flow) =
 MODIS_scan_radiances
 + MODIS_scan_level_metadata
 + MODIS_scan_pixel_quality
 + Decommuted_eng_mem_list

MODIS_scan_radiances (data flow) =
 [Half_frame_daymode_data
 | Frame_nightmode_data]

Half_frame_daymode_data (data flow) =
 5 { 2 { 16 { Unpacked_MODIS_radiance } 16
 } 2
 + 5 { 4 { Unpacked_MODIS_radiance } 4
 } 5
 + 14 { Unpacked_MODIS_radiance } 14
 + 17 { Unpacked_MODIS_radiance } 17
 } 5

Unpacked_MODIS_radiance (data flow, del) = *

Type: Integer (16-bit)

Range: -32768 - 32767

Description: This is the instantaneous output of a MODIS detector, unpacked into 16 bits. *

Frame_nightmode_data (data flow) =
 10 { 17 { Unpacked_MODIS_radiance } 17
 } 10

Unpacked_MODIS_radiance (data flow, del) = *

Type: Integer (16-bit)

Range: -32768 - 32767

Description: This is the instantaneous output of a MODIS detector, unpacked into 16 bits. *

MODIS_scan_level_metadata (data flow) = *

Type: Structure

Range: N/A

Description: This is the structure which contains the various MODIS scan-level metadata variables referenced in the MODIS L1A product specification. *

MODIS_scan_pixel_quality (store, pel) = *

Type: Structure

Range: N/A

Description: This is the structure containing the pixel quality data for each scan. *

Decommutated_eng_mem_list (data flow, pel) = *

Type: Structure

Range: N/A

Description: This is the list of all engineering and memory data output by the instrument; the values represent the last values received for the respective engineering or memory data. *

Failed_Pkts (store) =
 {MODIS_pkt}

C.2.3 Level1A_Parameters

Level1A_Parameters (data flow) =
Spacecraft_indicator
+ Processing_mode

Spacecraft_indicator (data flow, del) = *

Type: Enumeration

Range: EOS-AM or EOS-PM

Description: This is the parameter which indicates which spacecraft's data is expected to be processed during this processing run. *

Processing_mode (data flow, del) = *

Type: Enumeration

Range: Normal or Expedited

Description: This is the parameter which indicates under which mode the current processing is run. *

C.2.4 Log_messages

Log_messages (data flow) =
Init_log_msgs
+ Packet_log_msgs
+ Scan_log_msgs
+ Product_log_msgs
+ Close_log_msgs

Init_log_msgs (data flow) =

Level0_open_log_msgs

+ Level0_header_log_msgs

Level0_open_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Open_Level0_file process. *

Level0_header_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Verify_Level0_file_header process. *

Packet_log_msgs (data flow) =

Read_packet_log_msgs

+ Verify_packet_log_msgs

+ Missing_packet_log_msgs

Read_packet_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Read_MODIS_packets process. *

Verify_packet_log_msgs (data flow) =

Verify_CCSDS_log_msgs

+ Verify_MODIS_log_msgs

+ Verify_checksum_log_msgs

Verify_CCSDS_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Verify_CCSDS_header_info process. *

Verify_MODIS_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Verify_MODIS_header_info process. *

Verify_checksum_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Verify_packet_checksum process. *

Packet_sequence_log_msgs (data flow) =

Convert_time_log_msgs

+ Compare_packet_states_log_msgs

Convert_time_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Convert_packet_time process. *

Compare_packet_states_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Compare_current_and_previous_packet_state process. *

Missing_packet_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Process_MODIS_packet process. *

Scan_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Create_MODIS_Scans processes. *

Product_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Create_Level_1A_Products processes. *

Close_log_msgs (data flow, pel) = *

Type: text

Range: N/A

Description: These are the status messages generated by the Close_processing_run processes. *

(This page intentionally left blank.)

APPENDIX D: GEOLOCATION DATA FLOW DIAGRAMS

**This EPS image does not contain a screen preview.
It will print correctly to a PostScript printer.**

File Name : Figure_D1.EPSF

Title : Untitled

Creator : Cadre Printing

CreationDate : Thu 11 Sep 1997 19:22:19 EDT

Pages : 0

Figure D-1. MODIS Geolocation

**This EPS image does not contain a screen preview.
It will print correctly to a PostScript printer.
File Name : Figure_D2.EPSF
Title : Untitled
Creator : Cadre Printing
CreationDate : Thu 11 Sep 1997 19:23:21 EDT
Pages : 0**

Figure D-2. Process Level 1A Data

This EPS image does not contain a screen preview.

It will print correctly to a PostScript printer.

File Name : Figure_D3.EPSF

Title : Untitled

Creator : Cadre Printing

CreationDate : Thu 11 Sep 1997 19:23:47 EDT

Pages : 0

Figure D-3. Compute Geolocation Fields

This EPS image does not contain a screen preview.

It will print correctly to a PostScript printer.

File Name : Figure_D4.EPSF

Title : Untitled

Creator : Cadre Printing

CreationDate : Thu 11 Sep 1997 19:24:20 EDT

Pages : 0

Figure D-4. Determine Instrument Lines of Sight

**This EPS image does not contain a screen preview.
It will print correctly to a PostScript printer.**

File Name : Figure_D5.EPSF

Title : Untitled

Creator : Cadre Printing

CreationDate : Thu 11 Sep 1997 19:24:45 EDT

Pages : 0

Figure D-5. Determine Observed Locations

This EPS image does not contain a screen preview.

It will print correctly to a PostScript printer.

File Name : Figure_D6.EPSF

Title : Untitled

Creator : Cadre Printing

CreationDate : Thu 11 Sep 1997 19:25:12 EDT

Pages : 0

Figure D-6. Correct Observed Locations

This EPS image does not contain a screen preview.

It will print correctly to a PostScript printer.

File Name : Figure_D7.EPSF

Title : Untitled

Creator : Cadre Printing

CreationDate : Thu 11 Sep 1997 19:25:43 EDT

Pages : 0

Figure D-7. Compute Sensor and Solar Vectors

**This EPS image does not contain a screen preview.
It will print correctly to a PostScript printer.**

File Name : Figure_D8.EPSF

Title : Untitled

Creator : Cadre Printing

CreationDate : Thu 11 Sep 1997 19:26:23 EDT

Pages : 0

Figure D-8. Compute Solar Diffuser Angles

**This EPS image does not contain a screen preview.
It will print correctly to a PostScript printer.
File Name : Figure_D9.EPSF
Title : Untitled
Creator : Cadre Printing
CreationDate : Thu 11 Sep 1997 19:27:45 EDT
Pages : 0**

Figure D-9. Compute Lunar Vectors

**This EPS image does not contain a screen preview.
It will print correctly to a PostScript printer.
File Name : Figure_D10.EPSF
Title : Untitled
Creator : Cadre Printing
CreationDate : Thu 11 Sep 1997 19:28:17 EDT
Pages : 0**

Figure D-10. Write Geolocation Product

**This EPS image does not contain a screen preview.
It will print correctly to a PostScript printer.
File Name : Figure_D11.EPSF
Title : Untitled
Creator : Cadre Printing
CreationDate : Thu 11 Sep 1997 19:28:46 EDT
Pages : 0**

Figure D-11. Assess Geolocation Accuracy

(This page intentionally left blank.)

APPENDIX E: GEOLOCATION DATA DICTIONARY

This appendix contains the data dictionary elements that support the Geolocation requirements model. The appendix is divided into two sections.

E.1 Alphabetic Listing

This section is an alphabetic listing of all elements within the data dictionary. If a data dictionary entry is a composite of other data dictionary entries, then the entry shows the relationships to the other data dictionary entries using the notation found in the Glossary/Symbols section. If a data dictionary entry is not composed of other data dictionary entries, then that data dictionary entry shows the entry's type, range, and description, and the entry is labeled as either cel, del, or pel (see Glossary/Symbols section for explanation).

band_position (data flow, cel) = *

Type: real

Range: -14 - 15

Description: Area coverage for granule as defined in B.0 Earth Science Data Model, 420-TP-015-001, for the BoundingBox metadata class. *

Bounding Rectangle (data flow, cel) = *

Type: structure

Range: N/A

Description: Array of detector along-scan offsets on focal planes from ideal band, one offset per band. *

computation_status_messages (data flow, pel) = *

Type: text

Range: N/A

Description: Status messages generated during computation of geolocation fields. *

control_point_exclusion_mask (data flow, pel) = *

Type: structure

Range: N/A

Description: Data identifying MODIS scan regions that are not usable for control point analysis. *

Control_Point_Inventory_Metadata (data flow) =

Type: structure

Range: N/A

Description: Limited ECS core metadata for non-product files consisting of ShortName and Version ID. *

Control_Point_Matchups (data flow) =

land_control_point_matchups

+ ocean_control_point_matchups

control_point_metadata (data flow) =
control_point_specific_metadata
+ control_point_Inventory_Metadata

control_point_parameters (data flow) =
land_control_point_parameters
+ ocean_control_point_parameters

control_point_specific_metadata (data flow) =

Type: text

Range: N/A

Description: Granule level metadata stored in the Control Point product consisting of: Spacecraft ID, Production Date, Granule Number, Range Beginning Date, Range Beginning Time, Range Ending Date, Range Ending Time, s/c start position, s/c end position, Orbit number, Parameter file version, Geo parameter file version, Band number (band used for land control point correlation), cloud select, Snow select, Ice select, Number of Records. *

control_point_radiances (data flow) =

land_control_point_radiances

+ocean_control_point_radiances

Control_Points (data flow) =

land_control_points

+ ocean_control_points

converted_instrument_scan_data (data flow) =

scan_number

+ scan_start_times

+ mirror_sides

+ number_of_Earth_view_frames

+ scan_mirror_start_positions

+ converted_mirror_encoder_data

+ solar_diffuser_start_times

+ number_of_solar_diffuser_frames

+ space_view_start_times

+ number_of_space_view_frames

+ scan_mirror_channel_data

converted_mirror_encoder_data (data flow, del) = *

Type: integer

Range: 18000-18100 microseconds

Description: Array of mirror encoder measurements for each scan converted to incremental times. *

converted_spacecraft_ancillary_data (data flow) =
spacecraft_time_tags
+ spacecraft_orbit_position_vector
+ spacecraft_orbit_velocity_vector
+ spacecraft_attitude_angles

core_granule_processing_metadata (data flow) =*
Type: structure
Range: N/A
Description: Granule processing ECS Core Metadata listed in the "MODIS V2 Software Delivery Guide", Section 3.6.3, for Level 1 products.*

CP_calibrated_location (data flow) =*
Type: real
Range: N/A
Description: ECR coordinates of the true location of the control point.*

CP_Error_Flag (data flow) =*
Type: bit flags
Range: N/A
Description: Indicators for the following possible errors: failure to reach control point correlation threshold, control point may be outside of search window, cloudy scene rejected, multiple possible control point matches*.

CP_frame_number (data flow, cel) =*
Type: real
Range: N/A
Description: Fractional (1km) frame number where the control point is observed.*

CP_line_number (data flow, cel) =*
Type: real
Range: N/A
Description: Fractional (1km) line number where the control point is observed.*

CP_observation_time (data flow) =*
Type: real
Range: N/A
Description: TAI time of control point observation

CP_observed_location (data flow) =*

Type: real

Range: N/A

Description: ECR coordinates of the matchup algorithm's observed location of the control point.*

CP_orbit_position (data flow) =*

Type: real

Range: -7100000 - 7100000 meters

Description: The spacecraft orbit position in the ECR reference frame at the time of the control point observation.*

CP_orbit_velocity (data flow, cel) =*

Type: real

Range: -7520 - 7520 meters/second

Definition: The spacecraft orbit velocity in the ECR (non-inertial) reference-frame at the time of the control point observation.*

CP_spacecraft_attitude_angles (data flow) =*

Type: real

Range: -.01 - .01

Description: Spacecraft roll, pitch, and yaw angles at time of control point observaion.*

CP_view_vector (data flow) =*

Type: real

Range: -1.0 - 1.0

Description: View vector pointing to the observed control point, in the spacecraft coordinate system.*

data_input_status_messages (data flow, pel) = *

Type: text

Range: N/A

Description: Status messages generate during reading and processing of input data. *

data_output_status_messages (data flow, pel) = *

Type: text

Range: N/A

Description: Status messages generated during writing of output data to product file. *

detector_offsets (data flow, cel) = *

Type: real

Range: -.0001 - .0001 meters

Description: Size number_of_bands+_1 x 2 array of offsets of detector positions from nominal locations on focal plane. *

detector_space (data flow, cel) = *

Type: real

Range: 0.00013 - .00055

Description: Size (number of bands + 1) array of nominal detector spacing on the focal plane in the cross-scan direction. *

digital_elevation_model_data (data flow, pel) =*

Type: structure

Range: N/A

Description: Elevation data from the ECS 1km Digital Elevation Model.*

digital_elevation_model_metadata (data flow, pel) =*

Type: structure

Range: N/A

Description: Metadata indicating Elevation_Data of inferior quality from the ECS 1km Digital Elevation Model.*

Earth_View_center_data (data flow) =*

Earth_View_center_time

+ ECR_orbit_position

+ ECR_orbit_velocity

+ EVC_spacecraft_attitude_angles

+ECR_sun_ref

Earth_View_center_time (data flow, cel) =*

Type: real

Range: N/A

Description: spacecraft ephemeris time for the center of the Earth View scan*.

ECR_orbit_position (data flow, cel) =*

Type: real

Range: -7100000 - 7100000 meters

Description: The spacecraft orbit position in the ECR reference frame for the center of the Earth View scan.*

ECR_orbit_velocity (data flow, cel) =*

Type: real

Range: -7520 - 7520 meters/second

Definition: The spacecraft orbit velocity in the ECR (non-inertial) reference-frame for the center of the Earth View scan.*

ECR_sun_ref (data flow, pel) =*

Type: real

Range: -1 - 1

Description: Size 3 array of the sun_reference_vector in the ECR frame for the center of the Earth View scan*.

ECS_archive_metadata (data flow) =*

LocalInputGranuleID
+ BoundingBox

ECS_core_metadata (data flow) =

granule_ID_metadata
+ temporal_metadata
+ spatial_metadata
+ quality_metadata
+ core_granule_processing_metadata
+ software_version_number
+ input_files
+ granule_level_state

Elevation_Data (data flow) =

digital_elevation_model_metadata
+ digital_elevation_model_data

ellipsoid_intersection (data flow, cel) = *

Type: real

Range: -6400000 - 6400000 meters

Description: Size {N x 3} array representing N geocentric vectors from Earth center to observed locations in the ECR reference frame. *

encoder_time (data flow, cel) = *

Type: real

Range: 0.000001 second

Description: Units of encoder time measurements (1 microsecond). *

Ephemeris_source (data flow, pel) =*

Type: string

Range: 'spacecraft_ancillary data' 'Spacecraft_Data'

Description: Process control parameter to select the source of ephemeris and attitude information that is to be used to perform geolocation.*.

estimated_rms_error (data flow, cel) =*

Type: real

Range: -1,0 - 10000 meters

Description: Estimated geolocation rms error.*.

EVC_spacecraft_attitude_angles (data flow, cel) =*

Type: real

Range: -.01 - .01

Description: Spacecraft roll, pitch and yaw angles for the center of the Earth View scan.*.

first_sample_offset (data flow, cel) = *

Type: real

Range: 0 - .75

Description: Array of time offsets of first sample for a band to time of 1 km frame; units of frame_sample_time; nominally zero for all except 500m and 250m bands. *

focal_length (data flow, cel) = *

Type: real

Range: 0.280 - 0.381 meters

Description: Array focal lengths corresponding to each MODIS band. All lengths for a given focal plane will have same value. *

focal_plane_geometry (data flow) =

number_of_samples

+ band_position

+ detector_space

+ detector_offsets

+ first_sample_offset

+ focal_length

frame_sample_time (data flow, cel) = *

Type: real

Range: 0.000333 - 0.000334 second

Description: Frame sample interval for 1 km bands (3000 hz) *

geodetic_latitude (data flow, cel) =*

Type: double

Range: -90 - 90

Description: Size N array of geodetic latitudes for N observed locations*

geodetic_longitude (data flow, cel) =*

Type: double

Range: -180 - 180

Description: Size N array of geodetic longitudes for N observed locations*

Geolocation_Data (data flow) =

geolocation_global_metadata

+ instrument_parameters

+ geolocation_scan_data

geolocation_global_metadata (data flow) =

ECS_core_metadata

+ ECS_archive_metadata

+ product_specific_metadata

geolocation_parameter_metadata (data flow) =

parameter_file_version

+ estimated_rms_error

Geolocation_Parameters (data flow) =
instrument_parameters
+ level1a_conversion_and_validation_parameters
+ control_point_parameters
+ geolocation_parameter_metadata

geolocation_scan_data (data flow) =
product_fields_per_scan
+ product_fields_per_spatial_element

geolocation_scan_metadata (data flow) =
solar_diffuser_angles
+ lunar_vector
+ Earth_View_Center_data
+ scan_quality_flags

granule_ID_metadata (data flow) =*
Type: structure
Range: N/A
Description: Granule ID ECS core metadata listed in the "MODIS V2 Software Delivery Guide," Section 3.6.3, for Level 1 products.*

granule_level_state (data flow) =
granule_science_state
+ granule_science_abnormal

granule_metadata (data flow, pel) = *
Type: structure
Range: N/A
Description: Granule product-specific metadata. *

granule_science_abnormal (data flow, del) =*
Type: integer
Range: 0 1
Description: ECS searchable (Inventory) metadata flag that indicates that science_abnormal was set for at least one scan in the granule.*.

granule_science_state (data flow, del) =*
Type: integer
Range 0 1
Description: ECS searchable (Inventory) metadata flag that indicates that science_state was set for at least one scan in the granule.*.

height (data flow, cel) = *
Type: double
Range: 0 - 10,000
Description: size N array of heights of observed locations above reference ellipsoid (WGS84 standard). *

initiation_messages (data flow, pel) = *

Type: text

Range: N/A

Description: Message generated by the commencement of a program. *

input_files (data flow, pel) =*

Type: structure

Range: N/A

Description: Granule input files metadata listed in the "MODIS V2 Software Delivery Guide," Section 3.6.3, for Level 1 products.*

instrument_data_conversion_parameters (data flow, pel) = *

Type: structure

Range: N/A

Description: Locations and conversion parameters for mirror encoder times and sector start data in MODIS engineering data as defined in CDRL 305. *

instrument_data_validation_parameters (data flow, pel) = *

Type: structure

Range: N/A

Description: Set of parameters used to detect corrupted mirror encoder values. *

instrument_parameters (data flow) =

instrument_timing

+ mirror_model

+ focal_plane_geometry

+ internal_coordinate_transformations

instrument_scan_data (data flow) =

scan_start_times

+ mirror_sides

+ number_of_Earth_view_frames

+ scan_mirror_start_positions

+ scan_mirror_encoder_data

+ solar_diffuser_start_times

+ number_of_solar_diffuser_frames

+ space_view_start_times

+ number_of_space_view_frames

+ scan_mirror_channel_data

+ scan_number

instrument_timing (data flow) =

frame_sample_time

+ vernier_time

+ encoder_time

+ N_reset

+ sample_impulse

instrument_to_solar_diffuser (data flow, cel) = *

Type: real

Range: -1 - 1

Description: 3 x 3 matrix representing transformation from instrument to solar diffuser coordinate systems. *

instrument_to_spacecraft (data flow, cel) = *

Type: real

Range: -1 - 1

Description: 3 x 3 matrix representing transformation from instrument to spacecraft coordinate systems. *

internal_coordinate_transformations (data flow) =

telescope_to_instrument

+ mirror_to_instrument

+ instrument_to_spacecraft

+ instrument_to_solar_diffuser

interpolated_spacecraft_data (data flow) =

spacecraft_orbit_position

+ spacecraft_orbit_velocity

+ spacecraft_attitude_angles

L1A_scan_quality_flags (data flow, pel) =*

Type: structure

Range: N/A

Description: Scan-level quality flags in the Level_1A_Data which are input to be included in the Geolocation_Data.*

land_control_point_bands (data flow, pel) = *

Type: Structure

Range: N/A

Description: Designation of MODIS bands to be used for land CP correlation. *

land_control_point_correlation_parameters (data flow, pel) = *

Type: Structure

Range: N/A

Description: Parameters which control the correlation of the land CPs with the MODIS radiance's. *

land_control_point_matchups (data flow) =*

Type: structure

Range: N/A

Description: Information collected for each land control point which was successfully correlated with MODIS data*

CP_observation_time

+ CP_orbit_position

+ CP_orbit_velocity

- + CP_spacecraft_attitude_angles
- + CP_line_number
- + CP_frame_number
- + CP_view_vector
- + CP_observed_location
- + CP_calibrated_location
- + CP_Error_Flag

land_control_point_parameters (data flow) =
land_control_point_bands
+ land_control_point_selection_parameters
+ land_control_point_correlation_parameters

land_control_point_radiances (data flow, pel) = *
Type: structure
Range: N/A
Description: MODIS radiance's for land_control_point_bands for regions surrounding selected_land_control_points *

land_control_point_selection_parameters (data flow, pel) = *
Type: Structure
Range: N/A
Description: Parameters which control the selection of land CPs for a granule. *

land_control_points (data flow, pel) = *
Type: structure
Range: N/A
Description: Land CP chips from library generated from high-resolution reference data for selected MODIS bands. *

land-sea_mask (data flow, del) = *
Type: integer
Range: 0-7
Description: EOS format land/sea data. *

land-sea_mask_at_observed_locations (data flow, del) = *
Type: integer
Range: 0-7
Description: EOS format land/sea data at each observed earth location. *

level1a_conversion_and_validation_parameters (data flow) =
spacecraft_ancillary_data_conversion_parameters
+ spacecraft_ancillary_data_validation_parameters
+ instrument_data_conversion_parameters
+ instrument_data_validation_parameters

level1a_global_metadata (data flow) =
ECS_core_metadata
+ granule_metadata

level1a_scan_flags (data flow) =

L1A_scan_quality_flags
+ science_abnormal
+ science_state

level1B_global_metadata (data flow) =

ECS_core_metadata
+ product_specific_metadata

level1B_scan_flags (data flow, pel) = *

Type: structure

Range: N/A

Description: Quality flags in the Level_1B_Data which are used to select control_point_radiances. *

level2_global_metadata (data flow) =

ECS_core_metadata
+ product_specific_metadata

level2_scan_flags (data flow, pel) = *

Type: structure

Range: N/A

Description: Quality flags in the level_2_Data which are used to select level_2_Classification_masks and other_level2_data. *

Level_1A_Data (data flow) =

spacecraft_ancillary_data
+ instrument_scan_data
+ level1a_global_metadata
+ level1a_scan_flags

Level_1B_Data (data flow) =

control_point_radiances
+ level1B_global_metadata
+ level1B_scan_flags

level_2_Classification_masks (data flow, pel) = *

Type: structure

Range: N/A

Description: MODIS Level 2 view classification (cloud) mask. *

Level_2_Data (data flow) =

level_2_Classification_masks
+ other_level2_data
+ level2_global_metadata
+ level2_scan_flags

lines_of_sight (data flow, cel) = *

Type: double

Range: -1 - 1

Description: Size N x 3 array containing N vectors which represent the observation lines of sight (LOS) in the instrument coordinate frame. *

lines_of_sight_ECR (data flow, cel) = *

Type: double

Range: -1 - 1

Description: Size N x 3 array containing N vectors which represent the observations LOS in the ECR reference frame. *

LocalInputGranuleID (data flow, pel) = *

Type: string

Range: N/A

Description: The MODIS product file name for the input product file. *

lunar_vector (data flow, cel) = *

Type: real

Range: -1 - 1

Description: Array containing the unit vector pointing to the Moon's center, in instrument coordinates. *

mirr_side1_range (data flow, pel) =*

Type: real

Range: -PI - 2PI

Description: Mirror angle range within which mirror side1 is used during Earth View.*

mirror_alpha_angle (data flow, cel) = *

Type: real

Range: -.00015 - .00015

Description: Mirror wedge angle in along-scan direction (see description in ATBD). *

mirror_angle_polynomial (data flow, cel) = *

Type: real

Range: N/A

Description: Array of polynomial coefficient for converting mirror encoder values to mirror angles. *

mirror_beta_angle (data flow, cel) = *

Type: real

Range: -.00015 - .00015

Description: Mirror wedge angle in cross-scan direction (see description in ATBD). *

mirror_encoder_positions (data flow, cel) = *

Type: real

Range: 0 - 16384

Description: Array of mirror encoder values which correspond to encoder time measurements. *

mirror_encoder_times (data flow, cel) = *

Type: real

Range: 0 - 0.5 seconds

Description: Array of times from start of Earth view (EV) sector of mirror encoder measurements. *

mirror_gamma_angle (data flow, cel) = *

Type: real

Range: -.00005 - .00005

Description: Mirror misalignment with rotation axis (see description in ATBD). *

mirror_model (data flow) =

mirror_alpha_angle

+ mirror_beta_angle

+ mirror_gamma_angle

+ mirror_angle_polynomial

+ mirr_side1_range

mirror_normal_vector (data flow, cel) = *

Type: double

Range: -1 - 1

Description: Size 3 array containing the mirror normal vector corresponding to the time of observation. *

mirror_scan_angle (data flow, cel) = *

Type: real

Range: 0 - 2pi

Description: Mirror scan angle corresponding to the time of observation. *

mirror_sides (data flow, del) = *

Type: Integer

Range: 0 or 1

Description: Mirror side index for the scan *

mirror_to_instrument (data flow, cel) = *

Type: real

Range: -1 - 1

Description: 3 x 3 matrix representing transformation from mirror to instrument coordinate systems. *

moon_reference_vector (data flow, cel) = *

Type: real

Range: -1 - 1

Description: Size 3 array containing the unit vector in the Moon direction in the ECR frame at the observation time. *

N_reset (data flow, del) =*

Type: integer

Range: 3

Description: Number of frame clock cycles from sector start to start of data collection*.

number_of_Earth_view_frames (data flow, del) = *

Type: integer

Range: 1 - 1354

Description: Number of frames of detector observation collected during the EV for each scan. *

number_of_samples (data flow, del) = *

Type: integer

Range: 1 - 4

Description: Array specifying number of samples per frame and detector for each band. *

number_of_solar_diffuser_frames (data flow, del) = *

Type: integer

Range: 1 - 1354

Description: Number of frames of detector observation collected during the solar diffuser view for each scan. *

number_of_space_view_frames (data flow, del) = *

Type: integer

Range: 1 - 1354

Description: Number of frames of detector observation collected during the space view for each scan. *

observed_locations (data flow) =

geodetic_latitude

+ geodetic_longitude

observed_locations_with_terrain_correction (data flow) =

geodetic_latitude

+ geodetic_longitude

+ height

ocean_control_point_bands (data flow, pel) = *

Type: Structure

Range: N/A

Description: Designation of MODIS bands to be used for ocean CP correlation. *

ocean_control_point_correlation_parameters (data flow, pel) = *

Type: Structure

Range: N/A

Description: Parameters which control the correlation of the ocean CPs with the MODIS radiance's. *

ocean_control_point_matchups (data flow, pel) = *

Type: structure

Range: N/A

Description: Information collected for each ocean CP which was successfully correlated with MODIS data. *

CP_observation_time

+ CP_orbit_position

+ CP_orbit_velocity

+ CP_spacecraft_attitude_angles

+ CP_line_number

+ CP_frame_number

+ CP_view_vector

+ CP_observed_location

+ CP_calibrated_location

+ CP_Error_Flag

ocean_control_point_parameters (data flow) =

ocean_control_point_bands

+ ocean_control_point_selection_parameters

+ ocean_control_point_correlation_parameters

ocean_control_point_radiances (data flow, pel) = *

Type: structure

Range: N/A

Description: MODIS radiance's for ocean_control_point_bands for regions surrounding selected_ocean_control_points. *

ocean_control_point_selection_parameters (data flow, pel) = *

Type: Structure

Range: N/A

Description: Parameters which control the selection of ocean CPs for a granule. *

ocean_control_points (data flow, pel) = *

Type: structure

Range: N/A

Description: Ocean CPs (islands) from library generated from high-resolution coastline database. *

other_level2_data (data flow, pel) = *

Type: structure

Range: N/A

Description: MODIS Level 2 measurements that can identify regions unusable for control point analysis (i.e. snow, sea-ice, etc.) *

parameter_file_version (data flow, pel) =*

Type: text

Range: N/A

Description: Version identifier of the file containing the Geolocation_Parameters used.*.

parameter_input_status_messages (data flow, pel) = *

Type: text

Range: N/A

Description: Status messages generate during reading and processing of geolocation parameters. *

process_control_parameters (data flow) =

Ephemeris_source

product_fields_per_scan (data flow) =

validated_instrument_data

+ geolocation_scan_metadata

product_fields_per_spatial_element (data flow) =

observed_locations_with_terrain_correction

+ sensor_and_solar_vectors

+ spatial_element_quality_flags

+ land-sea_mask_at_observed_locations

product_specific_metadata (data flow) =

+ granule_metadata

+ Ephemeris_source

QA_QC_Data (data flow) =

status_messages

QA_QC_Data_Store (store) =

QA_QC_Data

quality_metadata (data flow, pel) =*

Type: structure

Range: N/A

Description: Granule quality ECS core metadata listed in the "MODIS V2 Software Delivery Guide," Section 3.6.3, for Level 1 products.*

range (data flow, cel) = *

Type: real

Range: 0 - 1500000 meters

Description: Distance from observed location to sensor along LOS. *

sample_impulse (data flow) =*

Type: integer

Range: 100

Description: Number of mirror encoder pulses between each encoder sample.*

scan_mirror_channel_data (data flow, del) =*

Type: integer

Range: 0,1

Description: Scan mirror assembly channel selection (Channel A, Channel B)*

scan_mirror_encoder_data (data flow, del) = *

Type: integer

Range: 18000 - 18100 microseconds

Description: Array of raw encoder measurements which represent time differences between 100 encoder pulses across EV. *

scan_mirror_start_positions (data flow, del) = *

Type: integer

Range: 0 - 16384

Description: Array containing view sector start encoder values for each instrument view (see definition in CDRL 305). *

scan_number (data flow, del) =*

Type: integer

Range: 1 - 210

Description: Ordinal number label for a scan within a granule.*

scan_quality_flags (data flow, del) = *

Type: integer

Range: N/A

Description: Array of flags indicating quality of scan level data used for geolocation. Quality states include: 1. No valid mirror encoder data available for scan 2. L1A quality flags set for scan. *

scan_start_times (data flow, cel) = *

Type: real

Range: N/A

Description: Array of EV sector start times for each scan *

science_abnormal (data flow, del) =*

Type: integer

Range: 0 1

Description: L1A engineering data ground-set flag that indicates prior knowledge of potentially abnormal science data due to things other than MODIS, such as, maneuvers, data link, etc.*.

science_state (data flow, del) =*

Type: integer

Range 0 1

Description: L1A engineering data flag that indicates the Normal/Test configuration of the MODIS instrument*.

selected_land_control_points (data flow, pel) = *

Type: structure

Range: N/A

Description: Land CPs are selected for correlation processing from the CP library based on the geographic boundaries of the scene. *

selected_ocean_control_points (data flow, pel) = *

Type: structure

Range: N/A

Description: Ocean CPs are selected for correlation processing from the CP library based on the geographic boundaries of the scene. *

sensor_and_solar_vectors (data flow) =

sensor_vectors

+ solar_vectors

sensor_azimuth_angle (data flow) =*

Type: real

Range: -180 - 180

Description: Azimuth of vector from observed location to sensor, measured from local North toward East.*

sensor_vectors (data flow) =

sensor_azimuth_angle

+ sensor_zenith_angle

+ range

sensor_zenith_angle (data flow) =*

Type: real

Range: 0 - 90

Description: Angle between local vertical vector and vector from observed location to sensor*

software_version_number (data flow, pel) =*

Type: structure

Range: N/A

Description: Granule software version metadata listed in the "MODIS V2 Software Delivery Guide," Section 3.6.3, for Level 1 products.*

solar_azimuth_angle (data flow, cel) =*

Type: real

Range: -180 - 180

Description: Azimuth of Sun reference vector measured from local North toward East.*

solar_diffuser_angles (data flow, cel) =*

Type: real

Range: -180 - 180

Description: Array containing the azimuth and elevation angles of the Sun vector relative to the solar diffuser.*

solar_diffuser_start_times (data flow, cel) = *

Type: real

Range: N/A

Description: Array of solar diffuser view sector start times for each scan *

solar_diffuser_view_data (data flow) =

solar_diffuser_start_times

+ number_of_solar_diffuser_frames

solar_vectors (data flow) =

solar_azimuth_angle

+ solar_zenith_angle

solar_zenith_angle (data flow, cel) =*

Type: real

Range: 0 - 180

Description: Angle between local vertical vector and Sun reference vector.*

space_view_data (data flow) =

space_view_start_times

+ number_of_space_view_frames

space_view_start_times (data flow, cel) = *

Type: real

Range: N/A

Description: Array of space view sector start times for each scan *

spacecraft_ancillary_data (data flow, del) = *

Type: integer

Range: N/A

Description: Array of raw spacecraft ancillary data per scan as defined in GIIS. *

spacecraft_ancillary_data_conversion_parameters (data flow, pel) = *

Type: structure

Range: N/A

Description: Locations and conversion parameters for spacecraft ancillary data fields as specified in the EOS GIIS. *

spacecraft_ancillary_data_validation_parameters (data flow, pel) = *

Type: structure

Range: N/A

Description: Set of parameters used to detect corrupt spacecraft orbit and attitude data values. *

spacecraft_attitude_angles (data flow, cel) = *

Type: real

Range: -.01 - .01

Description: Size N x 3 array of spacecraft roll, pitch and yaw angles. *

Spacecraft_Data (data flow) =
spacecraft_time_tags
+ spacecraft_orbit_position
+ spacecraft_orbit_velocity
+ spacecraft_attitude_angles

spacecraft_orbit_position (data flow, cel) = *
Type: real
Range: -7100000 - 7100000 meters
Description: Size N x 3 array containing N vectors representing the spacecraft orbit position in the ECI reference frame. *

spacecraft_orbit_position_ECR (data flow, cel) = *
Type: real
Range: -7100000 - 7100000 meters
Description: Size N x 3 array containing N vector representing the spacecraft orbit position in the ECR reference frame. *

spacecraft_orbit_velocity (data flow, cel) = *
Type: real
Range: -7600 - 7600 meters per second
Description: Size N x 3 array containing N vectors representing the spacecraft orbit velocity in the ECI reference frame. *

spacecraft_orbit_velocity_ECR (data flow, cel) = *
Type: real
Range: -7520 - 7520 meters/second
Definition: Size N x 3 array containing N vectors representing the spacecraft orbit velocity in the ECR (non-inertial) reference-frame.*.

spacecraft_time_tags (data flow, cel) = *
Type: real
Range: N/A
Description: Time tags of spacecraft ancillary data samples. *

spatial_element_quality_flags (data flow, del) = *
Type: integer
Range: N/A
Description: Array of flags indicating quality of product_fields_per_spatial_element.
Quality states indicated include:
1. Insufficient or invalid input data
2. No ellipsoid intersection
3. DEM data of inferior quality
4. Invalid sensor range *

spatial_metadata (data flow, pel) =*

Type: structure

Range: N/A

Description: Granule spatial ECS core metadata listed in the "MODIS V2 Software Delivery Guide," Section 3.6.3, for Level 1 products.*

status_messages (data flow) =

parameter_input_status_messages

+ data_input_status_messages

+ computation_status_messages

+ data_output_status_messages

sun_reference_vector (data flow, cel) = *

Type: real

Range: -1 - 1

Description: Size 3 array containing the unit vector in the Sun direction in the ECR frame at the observation time. *

telescope_to_instrument (data flow, cel) = *

Type: real

Range: -1 - 1

Description: 3 x 3 matrix representing transformation from telescope to instrument coordinate systems. *

temporal_metadata (data flow) =*

Type: structure

Range: N/A

Description: Granule temporal ECS core metadata listed in the "MODIS V2 Software Delivery Guide," Section 3.6.3, for Level 1 products.*

termination_messages (data flow, pel) = *

Type: text

Range: N/A

Description: Message generated at the termination of a program. *

terrain_height (data flow, pel) = *

Type: real

Range: 0 - 10000 meters

Description: Array of terrain heights for a specified region. *

terrain_intersection (data flow, cel) = *

Type: real

Range: -6400000 - 6400000 meters

Description: Size N x 3 array representing N geocentric vectors from Earth center to observed locations in the ECR reference frame. *

time_of_observation (data flow, cel) = *

Type: real

Range: N/A

Description: Time that a MODIS data sample was collected, computed from the scan_start_times, frame number and instrument timing parameters. *

validated_earth_view_data (data flow) =

scan_start_times

+ mirror_sides

+ number_of_Earth_view_frames

+ validated_mirror_encoder_data

validated_instrument_data (data flow) =

validated_earth_view_data

+ solar_diffuser_view_data

+ space_view_data

+ scan_number

validated_level_1a_data (data flow) =

validated_instrument_data

+ validated_spacecraft_ancillary_data

+ scan_quality_flags

validated_mirror_encoder_data (data flow) =

mirror_encoder_positions

+ mirror_encoder_times

+ scan_mirror_channel_data

validated_spacecraft_ancillary_data (data flow) =

spacecraft_time_tags

+ spacecraft_orbit_position

+ spacecraft_orbit_velocity

+ spacecraft_attitude_angles

vector_from_observed_locations_to_spacecraft_position (data flow, cel) = *

Type: real

Range: -1600000 - 1600000 meters

Description: Array representing vector from observed location on Earth's surface to sensor (spacecraft) position in the ECR reference frame. *

vernier_time (data flow, cel) = *

Type: real

Range: 0.00000333 - 0.00000334

Description: Units of sector start vernier time adjustment (0.01 frame_sample_time). *

E.2 Major Structures Decomposed

This section of the data dictionary is organized by external entities shown on the model's context diagram (Figure 4-2). Each entity is listed with its complete decomposition. It is hoped that this alternate arrangement of data dictionary entries will provide a straightforward way for readers to understand the specific information contained in the external interfaces.

E.2.1 Level_1A_Data

Level_1A_Data (data flow) =

spacecraft_ancillary_data
 + instrument_scan_data
 + level1a_global_metadata
 + level1a_scan_flags

spacecraft_ancillary_data (data flow, del) = *

Type: integer

Range: N/A

Description: Array of raw spacecraft ancillary data per scan as defined in GIIS. *

instrument_scan_data (data flow) =

scan_start_times
 + mirror_sides
 + number_of_Earth_view_frames
 + scan_mirror_start_positions
 + scan_mirror_encoder_data
 + solar_diffuser_start_times
 + number_of_solar_diffuser_frames

scan_start_times (data flow, cel) = *

Type: real

Range: N/A

Description: Array of Earth view sector start times for each scan. *

mirror_sides (data flow, del) = *

Type: Integer

Range: 0 or 1

Description: Mirror side index for the scan. *

number_of_Earth_view_frames (data flow, del) = *

Type: integer

Range: 1 - 1354

Description: Number of frames of detector observation collected during the EV for each scan. *

scan_mirror_start_positions (data flow, del) = *

Type: integer

Range: 0 - 16384

Description: Array containing view sector start encoder values for each instrument view (see definition in CDRL 305). *

scan_mirror_encoder_data (data flow, del) = *

Type: integer

Range: 18000 - 18100 microseconds

Description: Array of raw encoder measurements which represent time differences between 100 encoder pulses across EV. *

solar_diffuser_start_times (data flow, del) = *

Type: real

Range: N/A

Description: Array of solar diffuser view sector start times for each scan. *

number_of_solar_diffuser_frames (data flow, del) = *

Type: integer

Range: 1 - 1354

Description: Number of frames of detector observation collected during the solar diffuser view for each scan. *

level1a_global_metadata (data flow) =

ECS_core_metadata

+ product_specific_metadata

ECS_core_metadata (data flow) =

granule_ID_metadata

+ project_ID_metadata

+ temporal_metadata

+ spatial_metadata

+ quality_metadata

granule_ID_metadata (data flow, pel) = *

Type: structure

Range: N/A

Description: Granule ID ECS core metadata. *

project_ID_metadata (data flow, pel) = *

Type: structure

Range: N/A

Description: Granule project ID ECS core metadata. *

temporal_metadata (data flow, pel) = *

Type: structure

Range: N/A

Description: Granule temporal ECS core metadata. *

spatial_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule spatial ECS core metadata. *

quality_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule quality ECS core metadata. *

product_specific_metadata (data flow) =
 input_files
 + software_version_number
 + granule_metadata

input_files (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule input files metadata. *

software_version_number (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule software version metadata. *

granule_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule product-specific metadata. *

level1a_scan_flags (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Scan-level quality flags in the Level_1A_Data which are input to be included in the Geolocation_Data. *

E.2.2 Geolocation Parameters

Geolocation_Parameters (data flow) =
 instrument_parameters
 + level1a_conversion_and_validation_parameters
 + control_point_parameters

instrument_parameters (data flow) =
 instrument_timing
 + mirror_model
 + focal_plane_geometry
 + internal_coordinate_transformations

instrument_timing (data flow) =
frame_sample_time
+ vernier_time
+ encoder_time

frame_sample_time (data flow, cel) = *
Type: real
Range: 0.000333 - 0.000334 second
Description: Frame sample interval for 1 km bands (3000 hz). *

vernier_time (data flow, cel) = *
Type: real
Range: 0.00000333 - 0.00000334
Description: Units of sector start vernier time adjustment (0.01 frame_sample_time). *

encoder_time (data flow, cel) = *
Type: real
Range: 0.000001 second
Description: Units of encoder time measurements (1 microsecond). *

mirror_model (data flow) =
mirror_alpha_angle
+ mirror_beta_angle
+ mirror_gamma_angle
+ mirror_angle_polynomial

mirror_alpha_angle (data flow, cel) = *
Type: real
Range: -.00015 - .00015
Description: Mirror wedge angle in along-scan direction (see description in ATBD). *

mirror_beta_angle (data flow, cel) = *
Type: real
Range: -.00015 - .00015
Description: Mirror wedge angle in cross-scan direction (see description in ATBD). *

mirror_gamma_angle (data flow, cel) = *
Type: real
Range: -.00005 - .00005
Description: Mirror misalignment with rotation axis (see description in ATBD). *

mirror_angle_polynomial (data flow, cel) = *

Type: real

Range: N/A

Description: Array of polynomial coefficient for converting mirror encoder values to mirror angles. *

focal_plane_geometry (data flow) =

number_of_samples

+ band_position

+ detector_space

+ detector_offsets

+ first_sample_offset

+ focal_length

number_of_samples (data flow, del) = *

Type: integer

Range: 1 - 4

Description: Array specifying number of samples per frame and detector for each band. *

band_position (data flow, cel) = *

Type: real

Range: -14 - 15

Description: Array of detector along-scan offsets on focal planes from ideal band, one offset per band. *

detector_space (data flow, cel) = *

Type: real

Range: 0.00013 - .00055

Description: Size (number of bands + 1) array of nominal detector spacing on the focal plane in the cross-scan direction. *

detector_offsets (data flow, cel) = *

Type: real

Range: -.0001 - .0001 meters

Description: Size number_of_bands_+_1 x 2 array of offsets of detector positions from nominal locations on focal plane. *

first_sample_offset (data flow) = *

Type: real

Range: 0 - .75

Description: Array of time offsets of first sample for a band to time of 1 km frame; units of frame_sample_time; nominally zero for all except 500m and 250m bands. *

focal_length (data flow, cel) = *

Type: real

Range: 0.280 - 0.381 meters

Description: Array focal lengths corresponding to each MODIS band. All lengths for a given focal plane will have same value. *

internal_coordinate_transformations (data flow) =

telescope_to_instrument

+ mirror_to_instrument

+ instrument_to_spacecraft

+ instrument_to_solar_diffuser

telescope_to_instrument (data flow, cel) = *

Type: real

Range: -1 - 1

Description: 3 x 3 matrix representing transformation from telescope to instrument coordinate systems. *

mirror_to_instrument (data flow, cel) = *

Type: real

Range: -1 - 1

Description: 3 x 3 matrix representing transformation from mirror to instrument coordinate systems. *

instrument_to_spacecraft (data flow, cel) = *

Type: real

Range: -1 - 1

Description: 3 x 3 matrix representing transformation from instrument to spacecraft coordinate systems. *

instrument_to_solar_diffuser (data flow, cel) = *

Type: real

Range: -1 - 1

Description: 3 x 3 matrix representing transformation from instrument to solar diffuser coordinate systems. *

E.2.3 level1a_conversion_and_validation_parameters

level1a_conversion_and_validation_parameters (data flow) =

spacecraft_ancillary_data_conversion_parameters

+ spacecraft_ancillary_data_validation_parameters

+ instrument_data_conversion_parameters

+ instrument_data_validation_parameters

spacecraft_ancillary_data_conversion_parameters (data flow, pel) = *

Type: structure

Range: N/A

Description: Locations and conversion parameters for spacecraft ancillary data fields as specified in the EOS GIIS. *

spacecraft_ancillary_data_validation_parameters (data flow, pel) = *

Type: structure

Range: N/A

Description: Set of parameters used to detect corrupt spacecraft orbit and attitude data values. *

instrument_data_conversion_parameters (data flow, pel) = *

Type: structure

Range: N/A

Description: Locations and conversion parameters for mirror encoder times and sector start data in MODIS engineering data as defined in CDRL 305. *

instrument_data_validation_parameters (data flow, pel) = *

Type: structure

Range: N/A

Description: Set of parameters used to detect corrupted mirror encoder values. *

E.2.4 control_point_parameters

control_point_parameters (data flow) =

land_control_point_parameters

+ ocean_control_point_parameters

land_control_point_parameters (data flow) =

land_control_point_bands

+ land_control_point_selection_parameters

+ land_control_point_correlation_parameters

land_control_point_bands (data flow, pel) = *

Type: Structure

Range: N/A

Description: Designation of MODIS bands to be used for land CP correlation. *

land_control_point_selection_parameters (data flow, pel) = *

Type: Structure

Range: N/A

Description: Parameters which control the selection of land Cps for a granule. *

land_control_point_correlation_parameters (data flow, pel) = *

Type: Structure

Range: N/A

Description: Parameters which control the correlation of the land CPs with the MODIS radiances. *

ocean_control_point_parameters (data flow) =

ocean_control_point_bands

- + ocean_control_point_selection_parameters
- + ocean_control_point_correlation_parameters

ocean_control_point_bands (data flow, pel) = *

Type: Structure

Range: N/A

Description: Designation of MODIS bands to be used for ocean CP correlation. *

ocean_control_point_selection_parameters (data flow, pel) = *

Type: Structure

Range: N/A

Description: Parameters which control the selection of ocean CPs for a granule. *

ocean_control_point_correlation_parameters (data flow, pel) = *

Type: Structure

Range: N/A

Description: Parameters which control the correlation of the ocean CP with the MODIS radiance's. *

E.2.5 Spacecraft Data

Spacecraft_Data (data flow) =

spacecraft_time_tags

+ spacecraft_orbit_position

+ spacecraft_orbit_velocity

+ spacecraft_attitude_angles

+ spacecraft_attitude_rates

spacecraft_time_tags (data flow, cel) = *

Type: real

Range: N/A

Description: Time tags of spacecraft ancillary data samples. *

spacecraft_orbit_position (data flow, cel) = *

Type: real

Range: -7100000 - 7100000 meters

Description: Size N x 3 array containing N vectors representing the spacecraft orbit position in the ECI reference frame. *

spacecraft_orbit_velocity (data flow, cel) = *

Type: real

Range: -7600 - 7600 meters per second

Description: Size N x 3 array containing N vectors representing the spacecraft orbit velocity in the ECI reference frame. *

spacecraft_attitude_angles (data flow, cel) = *

Type: real

Range: -.01 - .01

Description: Size N x 3 array of spacecraft roll, pitch and yaw angles. *

spacecraft_attitude_rates (data flow, cel) = *

Type: real

Range: -0.005 - 0.005 per second

Description: Size N x 43 array of spacecraft attitude rates. *

E.2.6 Elevation_Data

Elevation_Data (data flow) =
digital_elevation_model_metadata
+ digital_elevation_model_data

digital_elevation_model_data (data flow, pel) = *

Type: structure

Range: N/A

Description: Elevation data preprocessed to MODIS format. *

digital_elevation_model_metadata (data flow, pel) = *

Type: structure

Range: N/A

Description: Metadata for digital elevation model preprocessed to MODIS format. *

E.2.7 Control_Points

Control_Points (data flow) =
land_control_points
+ ocean_control_points

land_control_points (data flow, pel) = *

Type: structure

Range: N/A

Description: Land CP chips from library generated from high-resolution reference data for selected MODIS bands. *

ocean_control_points (data flow, pel) = *

Type: structure

Range: N/A

Description: Ocean CPs (islands) from library generated from high-resolution coastline database. *

E.2.8 Geolocation_Data

Geolocation_Data (data flow) =
 geolocation_global_metadata
 + instrument_parameters
 + geolocation_scan_data

geolocation_global_metadata (data flow) =
 ECS_core_metadata
 + product_specific_metadata

ECS_core_metadata (data flow) =
 granule_ID_metadata
 + project_ID_metadata
 + temporal_metadata
 + spatial_metadata
 + quality_metadata

granule_ID_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule ID ECS core metadata. *

project_ID_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule project ID ECS core metadata. *

temporal_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule temporal ECS core metadata. *

spatial_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule spatial ECS core metadata. *

quality_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule quality ECS core metadata. *

product_specific_metadata (data flow) =
 input_files
 + software_version_number
 + granule_metadata

input_files (data flow, pel) = *
Type: structure
Range: N/A
Description: Granule input files metadata. *

software_version_number (data flow, pel) = *
Type: structure
Range: N/A
Description: Granule software version metadata. *

granule_metadata (data flow, pel) = *
Type: structure
Range: N/A
Description: Granule product-specific metadata. *

instrument_parameters (data flow) =
instrument_timing
+ mirror_model
+ focal_plane_geometry
+ internal_coordinate_transformations

instrument_timing (data flow) =
frame_sample_time
+ vernier_time
+ encoder_time

frame_sample_time (data flow, cel) = *
Type: real
Range: 0.000333 - 0.000334 second
Description: Frame sample interval for 1 km bands (3000 hz).*

vernier_time (data flow, cel) = *
Type: real
Range: 0.00000333 - 0.00000334
Description: Units of sector start vernier time adjustment (0.01 frame_sample_time). *

encoder_time (data flow, cel) = *
Type: real
Range: 0.000001 second
Description: Units of encoder time measurements (1 microsecond). *

mirror_model (data flow) =
mirror_alpha_angle
+ mirror_beta_angle
+ mirror_gamma_angle
+ mirror_angle_polynomial

mirror_alpha_angle (data flow, cel) = *
Type: real
Range: -.00015 - .00015
Description: Mirror wedge angle in along-scan direction (see description in ATBD). *

mirror_beta_angle (data flow, cel) = *
Type: real
Range: -.00015 - .00015
Description: Mirror wedge angle in cross-scan direction (see description in ATBD). *

mirror_gamma_angle (data flow, cel) = *
Type: real
Range: -.00005 - .00005
Description: Mirror misalignment with rotation axis (see description in ATBD). *

mirror_angle_polynomial (data flow, cel) = *
Type: real
Range: N/A
Description: Array of polynomial coefficient for converting mirror encoder values to mirror angles. *

focal_plane_geometry (data flow) =
number_of_samples
+ band_position
+ detector_space
+ detector_offsets
+ first_sample_offset
+ focal_length

number_of_samples (data flow, del) = *
Type: integer
Range: 1 - 4
Description: Array specifying number of samples per frame and detector for each band. *

band_position (data flow, cel) = *
Type: real
Range: -14 - 15
Description: Array of detector along-scan offsets on focal planes from ideal band, one offset per band. *

detector_space (data flow, cel) = *

Type: real

Range: 0.00013 - .00055

Description: Size (number of bands + 1) array of nominal detector spacing on the focal plane in the cross-scan direction. *

detector_offsets (data flow, cel) = *

Type: real

Range: -.0001 - .0001 meters

Description: Size number_of_bands+_1 x 2 array of offsets of detector positions from nominal locations on focal plane. *

first_sample_offset (data flow, cel) = *

Type: real

Range: 0 - .75

Description: Array of time offsets of first sample for a band to time of 1 km frame; units of frame_sample_time; nominally zero for all except 500m and 250m bands. *

focal_length (data flow, cel) = *

Type: real

Range: 0.280 - 0.381 meters

Description: Array focal lengths corresponding to each MODIS band. All lengths for a given focal plane will have same value. *

internal_coordinate_transformations (data flow) =

telescope_to_instrument

+ mirror_to_instrument

+ instrument_to_spacecraft

+ instrument_to_solar_diffuser

telescope_to_instrument (data flow, cel) = *

Type: real

Range: -1 - 1

Description: 3 x 3 matrix representing transformation from telescope to instrument coordinate systems. *

mirror_to_instrument (data flow, cel) = *

Type: real

Range: -1 - 1

Description: 3 x 3 matrix representing transformation from mirror to instrument coordinate systems. *

instrument_to_spacecraft (data flow, cel) = *

Type: real

Range: -1 - 1

Description: 3 x 3 matrix representing transformation from instrument to spacecraft coordinate systems. *

instrument_to_solar_diffuser (data flow, cel) = *

Type: real

Range: -1 - 1

Description: 3 x 3 matrix representing transformation from instrument to solar diffuser coordinate systems. *

geolocation_scan_data (data flow) =

product_fields_per_scan

+ product_fields_per_spatial_element

product_fields_per_scan (data flow) =

validated_level_1a_data

+ geolocation_scan_metadata

validated_level_1a_data (data flow) =

validated_instrument_data

+ validated_spacecraft_ancillary_data

+ scan_quality_flags

validated_instrument_data (data flow) =

validated_earth_view_data

+ solar_diffuser_view_data

validated_earth_view_data (data flow) =

scan_start_times

+ mirror_sides

+ number_of_Earth_view_frames

+ validated_mirror_encoder_data

scan_start_times (data flow, cel) = *

Type: real

Range: N/A

Description: Array of Earth view sector start times for each scan. *

mirror_sides (data flow, del) = *

Type: Integer

Range: 0 or 1

Description: Mirror side index for the scan. *

number_of_Earth_view_frames (data flow, cel) = *

Type: integer

Range: 1 - 1354

Description: Number of frames of detector observation collected during the EV for each scan. *

validated_mirror_encoder_data (data flow) =

mirror_encoder_positions

+ mirror_encoder_times

mirror_encoder_positions (data flow, cel) = *

Type: real

Range: 0 - 16384

Description: Array of mirror encoder values which correspond to encoder time measurements. *

mirror_encoder_times (data flow, cel) = *

Type: real

Range: 0 - 0.5 seconds

Description: Array of times from start of EV sector of mirror encoder measurements. *

solar_diffuser_view_data (data flow) =

solar_diffuser_start_times

+ number_of_solar_diffuser_frames

solar_diffuser_start_times (data flow, cel) = *

Type: real

Range: N/A

Description: Array of solar diffuser view sector start times for each scan. *

number_of_solar_diffuser_frames (data flow, cel) = *

Type: integer

Range: 1 - 1354

Description: Number of frames of detector observation collected during the solar diffuser view for each scan. *

validated_spacecraft_ancillary_data (data flow) =

spacecraft_time_tags

+ spacecraft_orbit_position

+ spacecraft_orbit_velocity

+ spacecraft_attitude_angles

+ spacecraft_attitude_rates

spacecraft_time_tags (data flow, cel) = *

Type: real

Range: N/A

Description: Time tags of spacecraft ancillary data samples. *

spacecraft_orbit_position (data flow, cel) = *

Type: real

Range: -7100000 - 7100000 meters

Description: Size N x 3 array containing N vectors representing the spacecraft orbit position in the ECI reference frame. *

spacecraft_orbit_velocity (data flow, cel) = *

Type: real

Range: -7600 - 7600 meters per second

Description: Size N x 3 array containing N vectors representing the spacecraft orbit velocity in the ECI reference frame. *

spacecraft_attitude_angles (data flow, cel) = *

Type: real

Range: -.01 - .01

Description: Size N x 3 array of spacecraft roll, pitch and yaw angles. *

spacecraft_attitude_rates (data flow, cel) = *

Type: real

Range: -0.005 - 0.005 per second

Description: Size N x 43 array of spacecraft attitude rates. *

scan_quality_flags (data flow, del) = *

Type: integer

Range: N/A

Description: Array of flags indicating quality of scan level data used for geolocation. Quality states include: 1. No valid mirror encoder data available for scan 2. L1A quality flags set for scan. *

geolocation_scan_metadata (data flow) =

sun_reference_vector

+ solar_diffuser_angles

sun_reference_vector (data flow, cel) = *

Type: real

Range: -1 - 1

Description: Size 3 array containing the unit vector in the Sun direction in the ECR frame at the observation time. *

solar_diffuser_angles (data flow, cel) = *

Type: real

Range: -pi - pi

Description: Array containing the azimuth and elevation angles of the Sun vector relative to the solar diffuser. *

product_fields_per_spatial_element (data flow) =

observed_locations_with_terrain_correction + sensor_and_solar_vectors

+ spatial_element_quality_flags

observed_locations_with_terrain_correction (data flow) =

geodetic_latitude

+ geodetic_longitude

+ height

geodetic_latitude (data flow, cel) = *

Type: double

Range: $-\pi/2 - \pi/2$

Description: Size N array of geodetic latitudes for N observed locations. *

geodetic_longitude (data flow, cel) = *

Type: double

Range: $-\pi - \pi$

Description: Size N array of geodetic longitudes for N observed locations. *

height (data flow, cel) = *

Type: double

Range: 0 - 10,000

Description: size N array of heights of observed locations above reference ellipsoid (WGS84 standard). *

sensor_and_solar_vectors (data flow) =

sensor_vectors

+ solar_vectors

sensor_vectors (data flow) =

sensor_azimuth_angle

+ sensor_zenith_angle

+ range

sensor_azimuth_angle (data flow, cel) = *

Type: real

Range: $-\pi - \pi$

Description: Azimuth of vector from observed location to sensor, measured from local North toward East. *

sensor_zenith_angle (data flow, cel) = *

Type: real

Range: 0 - $\pi/2$

Description: Angle between local vertical vector and vector from observed location to sensor. *

range (data flow, cel) = *

Type: real

Range: 0 - 1500000 meters

Description: Distance from observed location to sensor along LOS. *

solar_vectors (data flow) =

solar_azimuth_angle

+ solar_zenith_angle

solar_azimuth_angle (data flow, cel) = *

Type: real

Range: -pi - pi

Description: Azimuth of Sun reference vector measured from local North toward East. *

solar_zenith_angle (data flow, cel) = *

Type: real

Range: 0 - pi

Description: Angle between local vertical vector and Sun reference vector. *

spatial_element_quality_flags (data flow, del) = *

Type: integer

Range: N/A

Description: Array of flags indicating quality of product_fields_per_spatial_element. Quality states indicated include:

1. Insufficient or invalid input data
2. No ellipsoid intersection
3. DEM data of inferior quality *

QA_QC_Data (data flow) =

status_messages

status_messages (data flow) =

parameter_input_status_messages +

data_input_status_messages +

computation_status_messages +

data_output_status_messages

parameter_input_status_messages (data flow, pel) = *

Type: text

Range: N/A

Description: Status messages generate during reading and processing of parameters. *

data_input_status_messages (data flow, pel) = *

Type: text

Range: N/A

Description: Status messages generate during reading and processing of input data. *

computation_status_messages (data flow, pel) = *

Type: text

Range: N/A

Description: Status messages generated during computation of geolocation fields. *

data_output_status_messages (data flow, pel) = *

Type: text

Range: N/A

Description: Status messages generated during writing of output data to product file. *

E.2.9 Control_Point_Matchups

Control_Point_Matchups (data flow) =

land_control_point_matchups

+ ocean_control_point_matchups

land_control_point_matchups (data flow, pel) = *

Type: structure

Range: N/A

Description: Information collected for each land CP which was successfully correlated with MODIS data. *

ocean_control_point_matchups (data flow, pel) = *

Type: structure

Range: N/A

Description: Information collected for each ocean CP which was successfully correlated with MODIS data. *

E.2.10 Level_1B_Data

Level_1B_Data (data flow) =

+ control_point_radiances

+ level1B_global_metadata

+ level1B_scan_flags

control_point_radiances (data flow) =

land_control_point_radiances

+ ocean_control_point_radiances

land_control_point_radiances (data flow, pel) = *

Type: structure

Range: N/A

Description: MODIS radiance's for land_control_point_bands for regions surrounding selected_land_control_points. *

ocean_control_point_radiances (data flow, pel) = *

Type: structure

Range: N/A

Description: MODIS radiance's for ocean_control_point_bands for regions surrounding selected_ocean_control_points. *

level1B_global_metadata (data flow) =
 ECS_core_metadata
 + product_specific_metadata

ECS_core_metadata (data flow) =
 granule_ID_metadata
 + project_ID_metadata
 + temporal_metadata
 + spatial_metadata
 + quality_metadata

granule_ID_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule ID ECS core metadata. *

project_ID_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule project ID ECS core metadata. *

temporal_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule temporal ECS core metadata. *

spatial_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule spatial ECS core metadata. *

quality_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule quality ECS core metadata. *

product_specific_metadata (data flow) =
 input_files
 + software_version_number
 + granule_metadata

input_files (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule input files metadata. *

software_version_number (data flow, pel) = *

Type: structure

Range: N/A

Description: Granule software version metadata. *

granule_metadata (data flow, pel) = *

Type: structure

Range: N/A

Description: Granule product-specific metadata. *

level1B_scan_flags (data flow, pel) = *

Type: structure

Range: N/A

Description: Quality flags in the Level_1B_Data which are used to select control_point_radiances. *

E.2.11 Level_2_Data

Level_2_Data (data flow) =

level_2_Data

+ other_level2_data

+ level1a_global_metadata

+ level1a_scan_flags

level_2_Classification_masks (data flow, pel) = *

Type: structure

Range: N/A

Description: MODIS Level 2 view classification (cloud) mask. *

other_level2_data (data flow, pel) = *

Type: structure

Range: N/A

Description: MODIS Level 2 measurements that can identify regions unusable for control point analysis (i.e. snow, sea-ice, etc.) *

level2_global_metadata (data flow) =

ECS_core_metadata

+ product_specific_metadata

ECS_core_metadata (data flow) =

granule_ID_metadata

+ project_ID_metadata

+ temporal_metadata

+ spatial_metadata

+ quality_metadata

granule_ID_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule ID ECS core metadata. *

project_ID_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule project ID ECS core metadata. *

temporal_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule temporal ECS core metadata. *

spatial_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule spatial ECS core metadata. *

quality_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule quality ECS core metadata. *

product_specific_metadata (data flow) =
 input_files
 + software_version_number
 + granule_metadata

input_files (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule input files metadata. *

software_version_number (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule software version metadata. *

granule_metadata (data flow, pel) = *
 Type: structure
 Range: N/A
 Description: Granule product-specific metadata. *

level2_scan_flags (data flow, pel) = *

Type: structure

Range: N/A

Description: Quality flags in the level_2_Data which are used to select level_2_Classification_masks and other_level2_data. *